

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Anderson Salles

**SERVIÇO WEB DE RECOMENDAÇÃO BASEADO EM
ONTOLOGIAS E GRAFOS PARA REPOSITÓRIOS
DIGITAIS**

Florianópolis

2017

Anderson Salles

**SERVIÇO WEB DE RECOMENDAÇÃO BASEADO EM
ONTOLOGIAS E GRAFOS PARA REPOSITÓRIOS
DIGITAIS**

Dissertação submetido ao Programa
de Pós-Graduação em Ciência da Com-
putação para a obtenção do Grau de
Mestre em Ciência da Computação.
Orientador
Universidade Federal de Santa Cata-
rina: Prof. Dr. Roberto Willrich

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Salles, Anderson

Serviço Web de Recomendação Baseado em Ontologias
e Grafos para Repositórios Digitais / Anderson
Salles ; orientador, Roberto Willrich - Florianópolis
SC, 2017.

90 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Centro Tecnológico, Programa de Pós
Graduação em Ciência da Computação.

Inclui referências.

1. Ciência da Computação. 2. Sistemas de
Recomendação. 3. Ontologias. 4. Grafos. 5.
Repositórios Digitais. I. Willrich, Roberto. II.
Universidade Federal de Santa Catarina. Programa de
Pós-Graduação em Ciência da Computação. III. Título.

Anderson Salles

**SERVIÇO WEB DE RECOMENDAÇÃO BASEADO EM
ONTOLOGIAS E GRAFOS PARA REPOSITÓRIOS
DIGITAIS**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Ciência da Computação”, e aprovado em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 03 de Março 2017.

Profa. Dra. Carina Friedrich Dorneles
Coordenador
Universidade Federal de Santa Catarina

Banca Examinadora:

Prof. Dr.Roberto Willrich
Universidade Federal de Santa Catarina

Prof. Dr. Rudinei Goularte
Universidade de São Paulo

Prof. Dr. Divino Ignacio Ribeiro Junior
Universidade do Estado de Santa Catarina

Prof. Dr.Renato Fileto
Universidade Federal de Santa Catarina

Prof. Dr.Ronaldo dos Santos Mello
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus queridos pais.

AGRADECIMENTOS

Ao meu orientador Dr. Roberto Willrich, pela constante ajuda e paciência durante todo o andamento do mestrado, à Universidade Federal de Santa Catarina bem como ao PPGCC, pela oportunidade e ao CNPq pelo suporte financeiro concedido.

Não sei onde eu to indo, mas sei que eu
to no meu caminho.

(Raul Seixas, 1977)

RESUMO

Repositórios Digitais (RDs) oferecem funcionalidades para gerenciar, armazenar e acessar conteúdos digitais de diversos tipos, como teses, dissertações, artigos científicos, vídeos, obras de arte e obras literárias. Com sua popularização, é crescente o número de conteúdos disponibilizados nos RDs, gerando o problema clássico da sobrecarga de informação. Uma solução muito usual atualmente para tratar este problema são os sistemas de recomendação, que oferecem aos usuários uma lista de itens que são potencialmente de interesse destes usuários. Todavia, nenhuma das soluções abertas de RDs oferecem atualmente funcionalidades de recomendação. Esta dissertação propõe um serviço de recomendação fracamente acoplado e multidomínio para RDs. O fraco acoplamento é possível graças ao oferecimento do sistema de recomendação na forma de *Web Service*. A flexibilidade de domínio é fornecida graças ao uso de ontologias para representar o conhecimento específico de domínio associado aos conteúdos dos RDs. A fim de tentar garantir o tempo de resposta do serviço, este trabalho adotou o uso de técnicas de recomendação baseadas em grafos, bem como o uso de um banco de dados orientado a grafos. A fim de testar a factibilidade da proposta, este trabalho apresenta dois usos de caso do sistema proposto em domínios diferentes.

Palavras-chave: Sistemas de Recomendação. Ontologias. Grafos. Repositórios Digitais.

ABSTRACT

Digital Repositories (RDs) offer functionality to manage, store and access digital content of various types, such as phd thesis, master's thesis, scientific works, videos and literary works. With its popularization, the number of contents made available in RDs is increasing, generating the classic problem of information overload. A solution currently adopted to address this problem is recommendation systems, which offer to users a list of items that are potentially of interest to these users. However, none of the open RD solutions currently offers recommendation functionality. This dissertation proposes a weakly coupled and multidomain recommendation service for RDs. Weak coupling is possible through a recommendation system developed as a Web Service. The domain flexibility is reached with the use of ontologies to represent the domain-specific knowledge associated with the RD content. In order to provide a good service response time, this work has adopted the use of graph-based recommendation techniques, as well as the use of a graph-oriented database. In order to test the feasibility of the proposal, this work presents two use cases of the proposed system in different domains.

Keywords: Recommendation Systems. Ontologies. Graphs. Digital Repositories.

LISTA DE FIGURAS

Figura 1	Elementos e estrutura do esquema conceitual de dados do IEEE LOM.....	30
Figura 2	Uso de Rds de código Aberto.	32
Figura 3	Recomendação híbrida.	46
Figura 4	Modelo de dados representado em grafo.....	47
Figura 5	<i>Web Service</i> de Recomendação Proposto	55
Figura 6	Ontologia RecOnt	58
Figura 7	Ontologia de Literatura	59
Figura 8	Ontologia de Literatura estendida	60
Figura 9	Grafo Resultante	65
Figura 10	Exemplo de modelagem do perfil dos usuários.....	66
Figura 11	Interface de Usuário da BDLB com a recomendação ...	78
Figura 12	Ontologia de Domínio do Caso de Uso DSpace.....	79
Figura 13	Ontologia de Contexto do Caso de Uso DSpace	80
Figura 14	Apresentação da recomendação no DSpace.	82
Figura 15	Tempo de resposta do Primeiro Experimento	83
Figura 16	Tempos de resposta do registro de acesso	85
Figura 17	Tempos de resposta da solicitação da recomendação ...	86

LISTA DE ABREVIATURAS E SIGLAS

RD	Repositórios digitais
BDLB	Biblioteca Digital de Literatura Brasileira
BDTB	Biblioteca Digital Brasileira de Teses e Dissertações
FC	Filtragem Colaborativa
FBC	Filtragem Baseada em Conteúdo
FH	Filtragem Híbrida
RecOnt	Recommendation Ontology
DCES	Dublin Core Metadata Element
LOM	Learning Object Metadata
OA	Objeto de Aprendizagem
IBICT	Instituto Brasileiro de Informação em Ciência e Tecnologia
OASISbr	Portal Brasileiro de Acesso Aberto à Informação Científica
CRUESP	Repositório da Produção Científica do Conselho dos Reitores das Universidades Estaduais Paulistas
JSPUI	Java Server Pages User Interface
XMLUI	eXtented Mark Language User Interface
SAX	Simple API for XML
LAMP	Linux, Apache, MySQL, PHP
KB	Knowledge Base
OWL	Web Ontology Language
JSPUI	Java Server Pages User Interface
XMLUI	eXtented Mark Language User Interface

SUMÁRIO

1	INTRODUÇÃO	21
1.1	PERGUNTA DA PESQUISA	24
1.2	JUSTIFICATIVA	24
1.3	OBJETIVO GERAL	25
1.4	OBJETIVOS ESPECÍFICOS	26
2	REPOSITÓRIOS DIGITAIS	27
2.1	DEFINIÇÃO DE REPOSITÓRIO DIGITAL	27
2.2	METADADOS	28
2.2.1	Iniciativa de Metadados Dublin Core	29
2.2.2	Metadados IEEE LOM	30
2.3	FEDERAÇÃO DE RDS	30
2.4	SOLUÇÕES ABERTAS DE RDS E DSPACE	32
2.4.1	DSpace	33
2.4.1.1	Tecnologias usadas	33
2.4.1.2	Arquitetura de sistema	34
2.4.1.3	Metadados	34
2.4.1.4	Handles	34
2.4.1.5	Interfaces	35
2.4.1.6	XMLUI: Aspectos e Temas	35
2.5	BDLB	36
2.6	SOBRECARGA DE INFORMAÇÃO	36
2.7	CONSIDERAÇÕES FINAIS	37
3	SISTEMAS DE RECOMENDAÇÃO	39
3.1	PERFIL DE USUÁRIOS	40
3.2	ESTRATÉGIAS DE RECOMENDAÇÃO PERSONALIZADAS	42
3.3	TÉCNICAS CLÁSSICAS DE RECOMENDAÇÃO	43
3.3.1	Filtragem Baseada em Conteúdo	43
3.3.2	Filtragem Colaborativa	44
3.3.3	Filtragem Híbrida	46
3.4	RECOMENDAÇÃO BASEADA EM GRAFOS	46
3.5	RECOMENDAÇÃO BASEADA EM ONTOLOGIAS	48
3.6	INTEGRAÇÃO DE SISTEMAS DE RECOMENDAÇÃO EM RDS	49
3.6.1	Frameworks de Recomendação	49
3.6.2	Ad-Ons de Recomendação para RDs	50
3.6.3	Serviço de Recomendação via Web Services	50

3.7	CONSIDERAÇÕES FINAIS	52
4	UM WEB SERVICE DE RECOMENDAÇÃO PARA RDS	53
4.1	VISÃO GERAL DO SERVIÇO PROPOSTO	54
4.2	MODELAGEM CONCEITUAL DOS DADOS	56
4.2.1	RecOnt: Ontologia de Recomendação	57
4.2.2	Ontologias de Domínio	58
4.2.3	Ontologias de Contexto	59
4.3	INTERFACE DO SERVIÇO	60
4.4	REPRESENTAÇÃO NA FORMA DE GRAFOS	63
4.5	ALGORITMOS DE RECOMENDAÇÃO	65
4.5.1	Técnica de Recomendação por popularidade baseada em grafo	66
4.5.2	Técnica de Recomendação por FBC baseada em grafos	66
4.5.3	Técnica de Recomendação por FC baseada em grafo	69
4.6	MUDANÇAS NECESSÁRIAS NO RD	70
5	IMPLEMENTAÇÕES E AVALIAÇÕES	73
5.1	PROTÓTIPO DO SERVIÇO DE RECOMENDAÇÃO ...	74
5.2	INTEGRAÇÃO DO SERVIÇO DE RECOMENDAÇÃO NA BDLB	74
5.2.1	Ontologia de Domínio e Ontologia de Contexto ...	75
5.2.2	Modificação no RD BDLB	75
5.3	INTEGRAÇÃO DO SERVIÇO DE RECOMENDAÇÃO NO DSPACE	78
5.3.1	Ontologia de Domínio e Ontologia de Contexto ...	78
5.3.2	Modificações no RD DSpace	80
5.4	AVALIAÇÕES DO SERVIÇO DE RECOMENDAÇÃO ..	82
5.4.1	Avaliação das Técnicas de Recomendação	83
5.4.2	Avaliação usando a BDLB	84
6	CONCLUSÃO	89
	REFERÊNCIAS	91
	ANEXO A – Ontologia RecOnt	101
	ANEXO B – Ontologia do Domínio de Literatura	105
	ANEXO C – Ontologia de Contexto de Literatura	109
	ANEXO D – Ontologia de Domínio do Experimento DSpace	113
	ANEXO E – Ontologia de Contexto do Experimento DSpace	117

1 INTRODUÇÃO

Repositórios Digitais (RDs) são sistemas, geralmente Web, que permitem a organização, busca e acesso a coleções de recursos digitais. Tais serviços muitas vezes se baseiam na existência de um conjunto de metadados que permitem descrever esses recursos (como título, autor e palavras-chave). Por sua vez, os recursos digitais podem ser de diferentes tipos de mídias (por exemplo, conteúdos textuais, imagens, vídeos e áudios) e, muitas vezes, tratando de domínios diferentes do conhecimento humano. A Biblioteca Digital de Literatura Brasileira (BDLB - <http://www.literaturabrasileira.ufsc.br>) é um exemplo de RD cujo domínio dos conteúdos é a literatura. Outro exemplo é a Biblioteca Digital Brasileira de Teses e Dissertações (BDTB - <http://bdtb.ibict.br>), que se trata na realidade de uma federação de RDs, onde a BDTB oferece um sistema único para busca em um conjunto de RDs de diferentes instituições de ensino.

Não cresce apenas o número de RDs disponíveis na Web, mas cresce também o tamanho das suas coleções. A título de exemplo, o Repositório Institucional da UFSC (<http://repositorio.ufsc.br>) conta atualmente com mais de 91500 itens, organizados em coleções dos mais diversos tipos, como teses, dissertações, trabalhos de conclusão de curso, relatórios técnicos, documentos administrativos e legais da UFSC. Apesar dos RDs oferecerem recursos para a organização dos conteúdos (com base em metadados) e ferramentas de busca, o aumento das coleções ocasiona o problema clássico de sobrecarga de informação, ou seja, muitas vezes o usuário realiza um grande esforço para encontrar conteúdos de seu interesse em um vasto conjunto de conteúdos.

Uma forma de apoiar os usuários de um RD na tarefa de localizar itens de seu interesse é adicionar ao RD um sistema de recomendação (KANTOR et al., 2011). Este tipo de sistema tem por objetivo identificar itens no RD que sejam considerados interessantes para o usuário, levando em conta seus interesses/preferências. Sistemas de Recomendação são uma subclasse de filtragem de informação, que visam remover informações irrelevantes ou redundantes de um conjunto grande de informações. As técnicas clássicas de filtragem de informação adotadas pelos sistemas de informação são a Filtragem Colaborativa (FC), a Filtragem Baseada em Conteúdo (FBC) e a Filtragem Híbrida (FH) que é uma combinação das duas anteriores (CAZELLA; NUNES, 2010).

Muitas técnicas de recomendação que se baseiam em avaliações

dos conteúdos são independentes do tipo de conteúdo, como é o caso geral da FC. Mas outras técnicas levam em consideração diferentes características do item a recomendar durante a filtragem de informação, características estas mantidas pelos RDs na forma de valores de metadados. Por exemplo, para recomendar artigos científicos, o assunto pode ser muito mais relevante que os autores dos artigos. Já em obras literárias, os autores das obras são muito relevantes para recomendar novas obras aos usuários.

Uma alternativa de adaptação do Sistema de Recomendação ao tipo de item a ser recomendado é representar o conhecimento relacionado a esses itens em bases de conhecimento ontológicas. Atualmente já existem algumas propostas de sistemas de recomendação baseadas em ontologias como os trabalhos de (SIEG; MOBASHER; BURKE, 2010) e (CANTADOR; CASTELLS; BELLOGÍN, 2011). A Maior parte destas técnicas consideram o uso de Ontologias de Domínio para caracterizar os itens a recomendar. Ontologias de Domínio descrevem conceitos, propriedades e relacionamentos existentes no domínio de conhecimento dos conteúdos a serem recomendados (BORST, 1997).

Além das técnicas de recomendação baseadas em ontologias, existem as técnicas baseadas em grafos ((ANJORIN et al., 2012), (DEMOVIC et al., 2013) e (HUANG et al., 2014)). Elas modelam na forma de grafos as informações levadas em consideração pelo sistema de recomendação. A meta de modelar as informações na forma de grafos é otimizar o processo de recomendação. Estas últimas tendem a ter tempos de respostas muito menores que soluções baseadas em ontologias, que dependem de soluções que ainda não alcançaram um desempenho aceitável no caso de grandes volumes de informações.

Muitas organizações tendem a utilizar implementações em código aberto de RDs, como DSpace (DSpace, 2015), EPrints (EPRINTS, 2016), Digital Commons (COMMONS, 2016) e Omeka (OMEKA, 2017). Outras implementam soluções próprias (como é o caso da BDLB), visando atender a necessidades específicas da organização. Mas infelizmente, em ambas as situações, os serviços de recomendação não estão geralmente disponíveis. Este trabalho considera que este baixo número de RDs com sistemas de recomendação deve-se à falta de alternativas abertas que sejam facilmente adaptáveis à solução adotada de RD e ao domínio de suas coleções.

Existem ao menos três alternativas que poderiam ser adotadas quando da implantação de um sistema de recomendação em RDs: uso de frameworks de recomendação ((OWEN et al., 2011), (EKSTRAND et al., 2011) e (GANTNER et al., 2011)); uso de plugins (ELLIOTT; RUTHER-

FORD; ERICKSON, 2008); e implantação de *Web services* de recomendação ((BEHAM et al., 2010), (REGO et al., 2013)). Como analisado mais adiante neste trabalho, o provimento do serviço de recomendação via *Web Service* é a melhor das três soluções citadas, pois ela é de mais fácil integração do Sistema de Recomendação em RDs abertos, pois, como demonstrado mais adiante nesta dissertação, ela exige o menor esforço de modificação do código do RD. Em outros termos, esta solução oferece um maior desacoplamento entre o sistema de recomendação e o RD. Outra vantagem da adoção de soluções baseadas em *Web Service* é a possibilidade de oferecer recomendação em RDs federados, onde um único serviço de recomendação pode atender a todos os usuários da federação, podendo recomendar qualquer item independente do RD onde ele esteja disponibilizado.

Sistemas de recomendação oferecidos na forma de *Web Service*, portanto, facilitam a integração deste serviço em soluções de RDs pré-existentes. Mas, muitas vezes, isso não é suficiente. Considera-se aqui que o sistema de recomendação seja facilmente adaptável também ao tipo de conteúdo disponibilizado no RD e ao domínio de conhecimento relacionado (como artigos científicos, músicas, filmes e obras literárias). Este último é importante caso o serviço de recomendação utilize técnicas que se baseiam em características dos conteúdos recomendados (e não apenas nas avaliações dos conteúdos pelos usuários).

Este trabalho propõe um serviço de recomendação seguindo uma arquitetura cliente-servidor para RDs. A solução adotada para a integração de RDs e o serviço de recomendação proposto é o *Rest Web Service* (GROUP, 2004) que oferece fraco acoplamento e com isso baixo impacto na alteração de código de soluções de RDs preexistentes.

Visando oferecer independência de domínio, o sistema proposto adota o uso de ontologias como forma de representação formalizada das informações sobre os usuários e conteúdos necessárias para a realização da filtragem de informação. Este trabalho propõe uma Ontologia de Recomendação (RecOnt) que especifica conceitos e propriedades relacionadas à tarefa de recomendação. Trata-se de uma Ontologia de Tarefa (*Task Ontology*), que segundo (GUARINO, 1998) descreve um vocabulário relacionado a uma tarefa genérica, no caso, uma tarefa de recomendação de itens.

Utilizando a RecOnt, administradores de RDs podem criar contextos de recomendação, identificando o RD e especificando conceitos que serão utilizados para descrever os itens do RD. O termo contexto é utilizado aqui para referenciar as circunstâncias em que se dará a recomendação, em particular, a identificação do RD e o domínio de

conhecimento relacionado com os itens disponibilizados pelo RD. Para a especificação dos conceitos usados na descrição de itens, os administradores devem reusar, ou especificar uma ontologia de domínio, além de estender esta ontologia de domínio com a RecOnt, para descrever o contexto de recomendação.

Além disso, a solução preserva a privacidade dos usuários, pois o serviço não mantém dados pessoais dos usuários, nem dos itens acessados. Para tal, o serviço manipula apenas URIs, cujas relações com os conteúdos e usuários são resolvidas internamente pelo RD. Após estabelecido o contexto de recomendação, o RD deve registrar via *Web Service* os acessos dos usuários aos itens e finalmente solicitar recomendações para a audiência selecionada (todos os usuários, um grupo específico ou um único usuário).

Com o objetivo de garantir reduzidos tempos de resposta, o serviço proposto adota técnicas de recomendação baseadas em grafos. Para possibilitar o uso de técnicas baseadas em grafos, este trabalho adotou uma solução de mapeamento do conhecimento ontológico em grafos.

Esta dissertação apresenta também uma implementação prova-de-conceito do *Web Service* proposto, que é utilizado em dois cenários de uso do serviço. O objetivo é avaliar a factibilidade do serviço e a adaptação a diferentes tipos de conteúdos da solução proposta, bem como analisar sua escalabilidade.

1.1 PERGUNTA DA PESQUISA

Este trabalho busca responder a pergunta a seguir: Como integrar de maneira simples e eficiente serviços de recomendação em diferentes soluções de RDs com diferentes tipos de conteúdos digitais sem prejudicar o tempo de processamento e/ou eficácia do método?

1.2 JUSTIFICATIVA

Várias técnicas de recomendação vêm sendo utilizadas e aprimoradas, tendo grande popularização principalmente nos sistemas de comércio eletrônico. A grande maioria dessas técnicas de recomendação são fortemente acopladas ao seu domínio e/ou sistema, de forma que características como reuso, conhecimento prévio em sistemas de recomendação e sobrecarga de processamento em certos casos são deixadas

de lado. Por exemplo, os trabalhos avaliados (DEMOVIC et al., 2013), (HUANG et al., 2014)), (SIEG; MOBASHER; BURKE, 2010) e (CANTADOR; CASTELLS; BELLOGÍN, 2011) incorporam o sistema de recomendação ao seu domínio.

No contexto de RDs, a grande maioria das organizações já adotam soluções abertas, como DSpace, EPrint e Omeka. Com o crescimento de número de itens das coleções disponíveis nos RDs, seria importante para os seus usuários o oferecimento de serviços de recomendação. Mas, infelizmente, não existem atualmente serviços de recomendação nativos em soluções abertas de RDs. Esta trabalho visa oferecer uma solução para facilmente integrar serviços de recomendação a RDs existentes, sem grandes esforços de alteração de código do RD. Só assim as instituições seriam motivadas a incorporar serviços de recomendação em seus RDs.

Além de oferecer uma solução global para as recomendações dos RDs, a técnica aqui proposta possibilita que o processamento e armazenamento das informações referentes à recomendação sejam feitas de forma desacoplada do sistema, substituindo-o por trocas de informações simples entre cliente e servidor.

1.3 OBJETIVO GERAL

Esta dissertação tem como objetivo propor uma solução de sistema de recomendação facilmente adaptável à plataformas de RDs existentes independente dos tipos de conteúdos oferecidos pelos RDs. Para alcançar estes objetivos, o serviço de recomendação é oferecido através de *Web Service*. E as informações relacionadas aos itens e usuários são especificados na forma de ontologias.

É importante observar que este trabalho não propõe novas técnicas de recomendação, mas uma solução simples e eficiente de integração de um sistema de recomendação a RDs preexistentes. Além disso, deve-se observar também que a proposta não se trata de um *Web Service* semântico, pois ontologias aqui são utilizadas simplesmente para especificar o conhecimento acerca dos itens.

O *Web Service* de recomendação proposto neste trabalho adota técnicas de recomendação baseadas em grafos, que, como visto mais adiante nesta dissertação, tem demonstrado resultados promissores. Para tal, esta proposta utiliza banco de dados orientados a grafos visando melhorar o desempenho das técnicas de recomendação baseadas em grafos.

A solução apresentada neste trabalho se estende não só a RDs, mas também a qualquer aplicação que tenha por interesse o desacoplamento do processamento de recomendações, tais como serviços de *streaming* e *e-commerce*. De toda forma, esta dissertação foca o uso da solução especificamente em RDs.

1.4 OBJETIVOS ESPECÍFICOS

Esta dissertação possui os seguintes objetivos específicos:

- Identificar, de forma global, conceitos gerais relacionados a sistemas de recomendação na forma de uma Ontologia de Recomendação;
- Definir uma forma de adaptar a técnica de recomendação ao domínio do conhecimento dos itens a recomendar utilizando a ontologia de Recomendação;
- Definir uma técnica de mapeamento de ontologias em grafos para o uso de técnicas de recomendação baseadas em grafos;
- Definir e documentar um *Web Service* de Recomendação;
- Definir algoritmos de recomendação baseados em grafos a partir de técnicas clássicas de recomendação;
- Implementar um protótipo prova-de-conceito do sistema de recomendação proposto;
- Demonstrar a factibilidade da proposta em relação ao aspecto de multidomínio e tempo de processamento.

O restante deste trabalho está organizado na forma que segue. O capítulo 2 apresenta conceitos gerais sobre repositório digitais, e o capítulo 3 aborda as principais características de sistemas de recomendação bem como, os trabalhos relacionados. O *Web Service* de recomendação proposto é apresentado no capítulo 4. Na sequência, o capítulo 5 apresenta o protótipo desenvolvido e dois estudos de caso. Finalmente, as conclusões são apresentadas no capítulo 6.

2 REPOSITÓRIOS DIGITAIS

O avanço tecnológico na área da Web, junto com a ampliação dos serviços de acesso banda larga, tem tornado o acesso à informação cada vez mais eficiente. É crescente o número de conteúdos digitais disponibilizados nas páginas Web. Estes conteúdos digitais são dos mais diversos tipos, incluindo documentos textuais, áudios e vídeos (CAZELLA; NUNES, 2010).

Muitos sistemas Web criam ambientes virtuais capazes de oferecer uma melhor experiência de busca e organização desses conteúdos. Os sistemas que oferecem formas de busca e organização de conteúdo digitais são comumente chamados de Repositórios Digitais (RDs). Geralmente, os RDs permitem a organização de conteúdos digitais na forma de coleções, geralmente de escopo específico. Este tipo de sistema apresenta uma alta adesão por universidades de todo mundo, visando a organização de seu material digital gerado ao longo dos anos.

Este capítulo revisa os principais conceitos relacionados a RDs. Na sequência, duas soluções de RDs são apresentadas, a solução aberta de RD DSpace (DSpace, 2015) e a BDLB (Biblioteca Digital de Literatura Brasileira), uma implementação de um RD para o domínio de literatura. Finalmente, este capítulo apresenta uma das problemáticas de RDs que motiva este trabalho.

2.1 DEFINIÇÃO DE REPOSITÓRIO DIGITAL

Baseado em (VIANA; ARELLANO; SHINTAKU, 2005), RD pode ser definido como um sistema capaz de prover o armazenamento de conteúdos digitais que tem a capacidade de manter e gerenciar estes conteúdos por longos períodos de tempo e prover o acesso apropriado. Em (HEERY; ANDERSON, 2005), os autores definem o termo “repositório” como uma coleção de objetos digitais, especializando o termo “Repositório Digital” para aqueles repositórios de coleções digitais contendo as seguintes características:

- O depósito do conteúdo no RD pode ser realizado pelo criador, proprietário ou qualquer outro usuário;
- O repositório gerencia os conteúdos e seus metadados;
- O RD deve oferecer um conjunto de serviços para seus usuários, como mecanismos de busca, depósito, recuperação e controle de

acesso dos conteúdos disponibilizados;

- O RD precisa ser confiável e bem gerenciado, as informações disponibilizadas devem ter veracidade de acordo com suas fontes e seu conteúdo bem organizado.

Existe uma certa confusão entre os termos Repositório Digital e Biblioteca Digital. (DUNCAN, 2003), apesar de afirmar que a utilização do termo Biblioteca se aplica bem ao repositório digital, define que uma Biblioteca Digital é onde os recursos estão armazenados e apenas os bibliotecários (ou profissionais da informação) mantêm o controle do que pode ser armazenado ou incluído na Biblioteca Digital. Já os Repositórios Digitais enfatizam mais o fato de que as pessoas podem contribuir com recursos adicionais que são distribuídos na comunidade.

Outro termo importante é o de Repositório Institucional. Trata-se de um tipo de RD que busca preservar e disseminar o conhecimento (em formato digital) de uma instituição ou comunidade universitária (CROW, 2002).

2.2 METADADOS

RDs adotam metadados para caracterização das coleções disponibilizadas. Os metadados são informações estruturadas que descrevem, explicam, localizam ou tornam mais fácil recuperar, usar ou gerenciar um recurso de informações. Metadados são muitas vezes chamados de dados sobre dados ou informações sobre informações (NISO, 2004). Em outros termos, metadados são dados que fornecem informações sobre um ou mais aspectos de dados como criador, data de criação, assunto, tipo, etc.

Segundo (FURTADO et al., 2009), no processo de recuperação da informação é possível aos usuários dos RDs realizarem consultas booleanas para especificar os valores de elementos de metadados desejados dos recursos que o usuário deseja recuperar.

Para melhorar a interoperabilidade entre os RDs, vários padrões de metadados vem sendo adotados e aprimorados. Os principais padrões na área são o Dublin Core (INITIATIVE, 2014) e IEEE LOM (IEEE-LTSC, 2004).

2.2.1 Iniciativa de Metadados Dublin Core

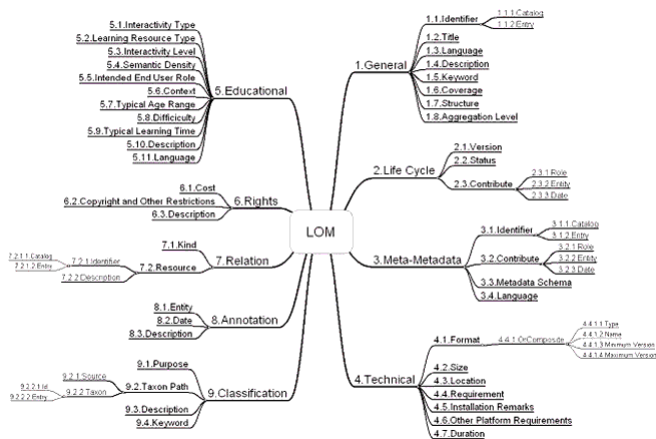
A Iniciativa de Metadados Dublin Core DCMI (INITIATIVE, 2014) é uma organização aberta que suporta a inovação compartilhada na área de metadados e melhores práticas em uma ampla gama de propósitos e modelos de negócio. A DCMI definiu o conjunto DCES (Dublin Core Metadata Element), um conjunto mínimo de 15 elementos capazes de descrever recursos digitais:

- *Title*: o título, ou nome, pelo qual o recurso é formalmente conhecido;
- *Creator*: Pessoa, organização ou serviço responsável pela criação do recurso;
- *Subject*: Assunto, palavras-chave, descritor ou código de classificação que descreve o tema do recurso;
- *Description*: conteúdo que descreve o conteúdo;
- *Publisher*: editor, incluindo pessoa, organização ou serviço;
- *Contributor*: pessoa, organização serviço que colaborou na criação do conteúdo;
- *Date*: data da criação ou disponibilização do recurso;
- *Type*: descrição do tipo do recurso, recomenda-se utilizar vocabulário controlado;
- *Format*: tipo da mídia ou as dimensões do recurso, pode ser usado para determinar o software, hardware ou outro equipamento necessário para processar o recurso;
- *Identifier*: identificador do recurso;
- *Source*: identificando a fonte que derivou o recurso (ou parte dele);
- *Language*: idioma do recurso;
- *Relation*: relação com outro recurso;
- *Coverage*: abrangência/cobertura, incluindo a localização espacial, período temporal ou jurisdição do recurso;
- *Rights*: gerenciamento de Direitos autorais.

2.2.2 Metadados IEEE LOM

O modelo de dados IEEE LOM (Learning Object Metadata) (IEEE-LTSC, 2004) é mais voltado ao contexto educacional, sendo um padrão que especifica a sintaxe e a semântica dos metadados visando descrever Objetos de Aprendizagem (OA). O IEEE define um OA como qualquer entidade, digital ou não, que possa ser usado para aprendizado, educação ou treinamento. A Figura 1 apresenta os elementos e estrutura do esquema conceitual de dados do IEEE LOM.

Figura 1 – Elementos e estrutura do esquema conceitual de dados do IEEE LOM.



Fonte: (IMS, 2002)

2.3 FEDERAÇÃO DE RDS

RDs podem ser unidos em Federações de RDs. O objetivo principal da criação de federação de RDs é oferecer aos seus usuários uma forma de oferecer um portal da federação para a busca sobre a federação de RDs (SAYÃO, 2007). A criação de federações de RDs exige que um grupo de organizações concorde que seus RDs estejam em conformidade com um conjunto de especificações, geralmente selecionadas a partir de padrões formalizados.

Uma forma muito usual atualmente para um RD abrir seus meta-

dados para o portal da federação é via o protocolo OAI-PMH (SAYÃO, 2007). Trata-se de um protocolo que segue o modelo Cliente-Servidor. O servidor OAI-PMH, chamado Provedor de Dados (*Data Provider*), é um RD que oferece uma interface na qual o cliente pode solicitar requisições OAI-PMH para acesso aos valores dos metadados do RD. O cliente OAI-PMH, chamado Provedor de Serviço (*Service Provider*), inicia requisições OAI-PMH para acesso aos metadados de um ou mais RDs. Este processo de consulta dos metadados é convencionalmente chamada de colheita de metadados (*harvesting*). Nas federações de RDs baseadas no protocolo OAI-PMH, os provedores de dados são os RDs fazendo parte da federação e o Provedor de Serviço oferece recursos para a implantação do Portal da federação.

No Brasil, existem algumas federações de RDs com diferentes propósitos, dentre elas pode-se citar:

- Biblioteca Digital Brasileira de Teses e Dissertações (BDTD): federação que agrega RDs de várias instituições de ensino superior do Brasil, incluindo a UFSC. Estes RDs disponibilizam as teses de doutorado e dissertações de mestrado destas instituições de ensino. O portal da federação é disponível via o URL <http://bdtb.ibict.br>. Esta ação é coordenada pelo IBICT (Instituto Brasileiro de Informação em Ciência e Tecnologia), que adotou o protocolo OAI-PMH e o esquema de metadados Dublin Core como padrões de interoperabilidade entre os RDs;
- Portal Brasileiro de Acesso Aberto à Informação Científica (OASISBr): federação que agrega RDs disponibilizando informações científicas, como os RDs da BDTD, RDs de periódicos e eventos científicos. O portal desta federação é acessível via O URL <http://oasis.ibict.br/>. Esta federação também é gerenciada pelo IBICT e adota os mesmos padrões da BDTD;
- Repositório da Produção Científica do Conselho dos Reitores das Universidades Estaduais Paulistas (CRUESP): federação restrita brasileira, disponível em <http://www.repositorio.cruesp.sp.gov.br/>, que visa preservar e proporcionar acesso aberto, público e integrado à produção científica de docentes, pesquisadores, alunos e servidores da USP (Universidade de São Paulo), Unicamp (Universidade Estadual de Campinas) e Unesp (Universidade Estadual Paulista).
- Federação de Repositórios Educa Brasil: Com o portal acessível via URL <http://feb.ufrgs.br/>, trata-se de um serviço Experi-

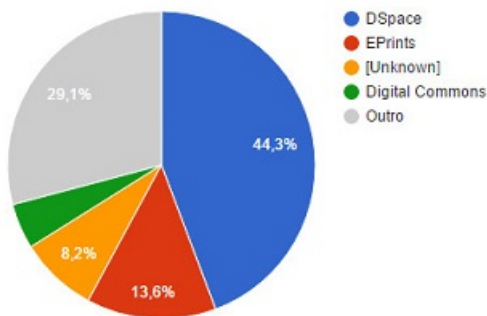
mental da RNP (Rede Nacional de ensino e Pesquisa) com o propósito de organizar diversos RDs disponibilizando objetos de aprendizagem (OA) visando o reuso destes OAs. Dezenas de instituições de ensino participam desta federação, incluindo a UFRGS (Universidade Federal do Rio Grande do Sul), UFSC, MEC (Ministério da Educação), Fiocruz e a Fundação Biblioteca Nacional.

2.4 SOLUÇÕES ABERTAS DE RDS E DSPACE

Para a implementação dos repositórios digitais, as mais diversas instituições optam por soluções abertas. Dentre as diversas soluções de RDs abertos disponíveis as que se destacam são: DSpace ((DSpace, 2015), Omeka (OMEKA, 2017), EPrints (EPRINTS, 2016), Digital Commons (COMMONS, 2016) e Fedora (FEDORA, 2017).

Segundo levantamento feito pelo OpenDOAR (Diretório de Repositórios de Acesso Aberto em inglês) o RD mais popular é o DSpace, atingindo um percentual próximo a 44%, como pode ser observado na Figura 2.

Figura 2 – Uso de Rds de código Aberto.



Fonte: <http://www.opendoar.org>

Obviamente, as organizações tendem a adotar soluções abertas para RDs, evitando os custos da aquisição ou desenvolvimento de soluções próprias da organização. Além disso, as organizações contam com a possibilidade de atualizações e manutenções constantes das soluções abertas, sem custo adicional de manter uma equipe de desenvolvimento própria.

Por ser a solução mais adotada, o DSpace foi escolhido para

o estudo de caso da proposta desta dissertação. A seção que segue introduz os principais aspectos desta solução aberta de RD.

2.4.1 DSpace

Criado em 2002 pelo MIT e os pelos laboratórios da Hewlett Packard (HP), é um software para criação de RDs, em geral RIs, registrando um total de 1356 implementações da plataforma, tornando-o assim o software mais usado mundialmente para tal finalidade. É usado por bibliotecas, arquivos e centros de pesquisa (CASTAGNÉ, 2013).

A plataforma DSpace possui uma estrutura baseada em comunidades e coleções, conseguindo assim, criar uma representação dos centros, departamentos ou unidades administrativas de uma instituição, podendo abrigar os mais variados formatos de mídias digitais (OLIVEIRA; CARVALHO, 2011).

Segundo os autores do projeto, DSpace é um software gratuito de âmbito acadêmico, sem fins lucrativos e para construção de repositórios digitais abertos para organizações. A plataforma preserva e permite acesso fácil e livre acesso para todos os tipos de conteúdos digitais, incluindo textos, imagens, imagens em movimento e conjuntos de dados. Possui uma crescente comunidade de desenvolvedores que ajudam a expandir e melhorar ainda mais o software para cada versão lançada (DSpace, 2015).

2.4.1.1 Tecnologias usadas

DSpace é uma solução de RD em código aberto (licença BSD), permitindo que qualquer pessoa, organização ou instituição possa criar seu RD de maneira totalmente gratuita. A linguagem de desenvolvimento utilizada foi JAVA e foram adotadas uma série de ferramentas, incluindo:

- Apache Maven: ferramenta que realiza a construção da árvore de pacotes necessários para instalação do DSpace, além da compilação do código fonte para instalação;
- Apache Ant: para a instalação do código compilado;
- Banco de dados relacional: podendo ser o PostgreSQL ou Oracle;
- Container Servlet 3.0: podendo ser o Apache Tomcat 7+ ou similar.

O DSpace pode ser instalado em máquinas Unix Linux, HP/UX, Mac OSX, etc.) ou Microsoft Windows.

2.4.1.2 Arquitetura de sistema

A arquitetura do DSpace é dividida em três camadas ((SMITH et al., 2003)):

- Camada de armazenamento: implementada usando um sistema de arquivos e gerenciada pelas tabelas do banco de dados (e.g., PostgreSQL);
- Camada de lógica de negócios: é onde as funcionalidades específicas do DSpace residem, incluindo o workflow, a gerência de conteúdo, a administração, módulos de busca e navegação;
- Camada de aplicação: cobre toda a interface do sistema.

2.4.1.3 Metadados

DSpace faz o uso do padrão Dublin Core para descrição de metadados de seus itens, havendo um mínimo de três campos requeridos para a composição de um item ((SMITH et al., 2003)): título, idioma e a data de submissão. Todos os outros campos, como resumo, palavras-chaves, detalhes técnicos, direitos autorais, entre outros, são opcionais. Os valores destes metadados podem ser visualizados através da interface Web de visualização de um item no DSpace.

2.4.1.4 Handles

Uma das metas do DSpace é possibilitar a preservação digital, permitindo que usuários possam localizar e baixar itens depositados a longo prazo. Para tal, é fundamental que itens possam ser referenciados em outros contextos/ambientes (como em artigos científicos) usando uma URL persistente que ofereça acesso ao item por longos períodos de tempo. Para tal, para o DSpace foi escolhida a solução CNRI Handles (<https://www.handle.net/>) como localizadores com persistência de cada item o qual é responsável pelas tarefas de gerência de itens dos repositórios cadastrados (SMITH et al., 2003).

2.4.1.5 Interfaces

O DSpace é um sistema Web oferecendo a seus usuários várias interfaces (SMITH et al., 2003): uma para submissão de itens e/ou qualquer processo que esteja envolvido com a submissão; uma para usuários finais que buscam pela informação; e outra para administradores do sistema.

A interface para usuários finais, ou interface pública, permite ao usuário navegar nas coleções e realizar buscas utilizando como critério os valores dos metadados. Uma vez que o usuário localize o item de interesse, ele pode acessar a página Web do item, onde são apresentados os detalhes do item (a partir dos valores dos metadados) e é possível realizar o *download* do conteúdo digital associado.

O Dspace implementa um fluxo de trabalho (*workflow*) para que submetedores possam depositar seus itens e avaliadores possam aceitar o recurso submetido, rejeitar, ou então encaminhar solicitações de revisão.

2.4.1.6 XMLUI: Aspectos e Temas

Ainda a respeito da interface Web, DSpace oferece duas opções de implementação, a JSPUI (Java Server Pages User Interface) e a XMLUI (eXtented Mark Language User Interface) . Cada uma destas opções tem suas vantagens e desvantagens e cabe ao administrador ou equipe responsável pelo repositório decidir qual delas utilizar (VIANA; ARELLANO; SHINTAKU, 2005).

A versão atualmente utilizada do XMLUI no DSpace é chamada de Manakin. Essa versão passa a fazer uso do framework Apache Cocoon(<http://cocoon.apache.org/>) para usar uma abordagem Simple API for XML (SAX) . Para completar essa customização, Manakin usa temas (*Themes*) para estilo de conteúdo e pacotes chamados de *Aspects* para modularizar a geração do conteúdo (SMITH et al., 2003). Em outros termos, *Aspects* são como extensões ou *plugins* para o Manakin. Eles são componentes interativos que modificam características existentes ou fornece novas características para o RD.

2.5 BDLB

A Biblioteca Digital de Literatura Brasileira (BDLB) (<http://www.literaturabrasileira.ufsc.br>) é um repositório digital que conta atualmente com uma vasta coleção de obras literárias de domínio público e informações sobre escritores brasileiros. Estão catalogados, no momento da escrita desta dissertação, mais de 74.000 documentos da área da literatura (obras literárias, manuscritos, cartas, etc.) e mais de 18.000 autores. Dos documentos catalogados, 4.646 estão disponíveis para acesso digital.

A BDLB é caracterizada pelo uso de código aberto, adotando a plataforma LAMP (Linux, Apache, MySQL, PHP). Existem diversos tipos de usuários: administradores, com a responsabilidade de configuração e personalização do sistema, bem como a atribuição de papéis aos usuários; colaboradores, com a função de cadastramento de dados sobre autores e obras literárias; e os usuários, que são alunos e professores de literatura, bem como leitores de obras literárias.

Para a descrição dos conteúdos da BDLB, além do título do conteúdo e autor, também há um conjunto de metadados descritores considerados relevantes para sua localização, como por exemplo Título, Autores, Gênero, Ano, Editora, entre outros.

Como critérios de busca a BDLB oferece várias estratégias como: Navegação por conteúdo, Busca simples, Busca por documentos e Busca por Autor.

Em termos de padrão de interoperabilidade, a BDLB adotou o esquema de metadados Dublin Core e o protocolo PMH-OAI.

2.6 SOBRECARGA DE INFORMAÇÃO

Como já citado, RDs oferecerem facilidades de organização dos conteúdos digitais em coleções e ferramentas de busca. De acordo com (WAN G. E LIU, 2000), três métodos de busca de informações são comuns em RDs:

- Navegação livre: onde o usuário navega por uma coleção para localizar uma informação específica;
- Busca textual: neste método, conteúdos baseados em textos são indexados para que o usuário possa fazer uso de termos ou palavras-chave. O critério de busca pode ser com base em valores de metadados ou então no próprio conteúdo de texto;

- Busca baseada em conteúdo: permite ao usuário definir critérios de busca baseados em características dos próprios conteúdos de imagens, áudios e vídeos. Algumas dessas características incluem cor, textura, tamanho ou movimentos.

Apesar dos RDs oferecerem recursos para a organização dos conteúdos e ferramentas de busca, o aumento das coleções ocasiona o problema clássico de sobrecarga de informação, ou seja, há um grande esforço do usuário para encontrar conteúdos de seu interesse em um grande conjunto de conteúdos disponíveis. A sobrecarga, no contexto deste trabalho, ocorre quando há uma quantidade de informações muito grande disponível numa busca, onde a quantidade de itens atendendo ao critério de busca é enorme, dificultando ao usuário encontrar o que estava procurando.

Um RD que sofre o problema de sobrecarga de informação é o repositório institucional da UFSC (<http://repositorio.ufsc.br>), que disponibiliza atualmente quase 92.000 recursos. Dependendo do critério de busca utilizado pelo usuário, centenas de itens podem ser retornados, dificultando a localização dos itens relevantes para o usuário.

Uma forma de apoiar os usuários dos RDs na localização de informações é a implantação de Sistemas de Recomendação. Atualmente, os RDs que fazem uso de sistemas de recomendação tendem a ter implementações próprias, que muitas vezes são dependentes de domínio e/ou audiência.

2.7 CONSIDERAÇÕES FINAIS

Neste capítulo foram vistos vários aspectos referentes aos RDs, como definições e apresentação do RD onde o sistema proposto está implementado.

Este capítulo também abordou o problema da sobrecarga de informação associado ao alto acoplamento de um sistema de recomendação. Um dos objetivos deste trabalho é a resolução deste problema, através do uso da recomendação e de técnicas que reduzam o espaço de busca do repositório e que desacoplem tanto o domínio quanto a audiência da recomendação, reduzindo assim o problema da sobrecarga de informações bem como o alto acoplamento de um sistema de recomendação.

O capítulo a seguir trata de sistemas de recomendação, abordando as principais técnicas, bem como os trabalhos relacionados à proposta apresentada nesta dissertação.

3 SISTEMAS DE RECOMENDAÇÃO

A quantidade de conteúdos digitais disponibilizados aos usuários via os diversos sistemas computacionais é imensa. Mesmo utilizando sistemas de busca, dependendo dos critérios utilizados, o número de itens que retornam em uma busca é ainda muito grande, dificultado o acesso a conteúdos relevantes ao usuário. Este problema é o problema conhecido como sobrecarga de informação.

A título ilustrativo, considere o trabalho de revisão sistemática de um dado tema de pesquisa. Neste processo, o pesquisador deve realizar buscas em diversas fontes disponibilizando artigos científicos. Em geral, a quantidade de trabalhos científicos disponibilizados é grande. Por exemplo, a Biblioteca Digital da ACM (ACM, 2017) possui mais de 400.000 artigos completos, a da IEEE (IEEE, 2017) possui mais de 4.000.000 de itens indexados, a Science Direct (SCIENCEDIRECT, 2017) oferece acesso a mais de 14.000.000 de publicações. Desta forma, dependendo dos critérios de busca definidos, este processo de revisão sistemática pode exigir uma sobrecarga de trabalho muito grande.

Uma das soluções frequentemente adotada atualmente para auxiliar os usuários no processo de localização de conteúdos de interesse de um usuário é o “Sistema de Recomendação”. Segundo (RESNICK; VARIAN, 1997), um sistema de Recomendação tem por finalidade auxiliar no processo social de indicar ou receber indicação, seja esta indicação referente a livros, artigos, discos, restaurantes ou notícias.

(TATIYA; VAIDYA, 2014) enquadra um sistema de recomendação no contexto de filtragem de informação. Segundo os autores, um Sistema de filtragem de informações é um sistema que remove informações irrelevantes ou desnecessárias antes de enviá-las para o usuário humano, de forma à filtrar a superabundância de dados na Web. Sistema de recomendação é considerado como uma subclasse de sistema de filtragem que prediz a ‘classificação’ ou ‘preferência’ que o usuário daria/tem sob um dado item.

Segundo (GOLDBERG; NICHOLSL, 1992), um dos primeiros sistemas de recomendação foi o Tapestry (GOLDBERG et al., 1992). Este sistema foi desenvolvido pela Xerox Palo Alto Research Center oferecendo funcionalidades de recomendação de *newsgroups* para os usuários. Desde então, sistemas de recomendação tornaram-se populares em uma variedade de aplicações práticas. Dentre elas, pode-se citar sistemas de recomendação de filmes, músicas, hotéis, notícias, livros, artigos científicos e produtos em geral.

Este capítulo tem como objetivo apresentar os principais conceitos relacionados a sistemas de recomendação. Dentre eles, serão apresentadas as técnicas clássicas de recomendação, bem como técnicas menos usuais que se baseiam em ontologias e grafos. Também serão analisadas propostas que se baseiam no uso de *Web Services*.

3.1 PERFIL DE USUÁRIOS

Os sistemas de recomendação podem ser divididos em dois grupos: sistemas de recomendação personalizados, e sistemas não personalizados. Um sistema de recomendação não personalizada são aqueles que não consideram qualquer tipo de informação sobre o usuário para gerar a recomendação. Desta forma, estes sistemas de recomendação, não são capazes de identificar as preferências e necessidades de longo termo de seus usuários nem de gerar essas recomendações automaticamente com base no perfil de cada usuário. Exemplos de sistemas de recomendação não personalizadas incluem a recomendação por popularidade e recomendações na área de comércio eletrônico do tipo "produtos similares e este produto".

Contrapondo os sistemas de recomendação não personalizados, os sistemas de recomendação personalizados são aqueles capazes de capturar dados individuais dos seus usuários e de gerar recomendações levando em consideração essas necessidades. Estes dados sobre os usuários podem incluir o histórico de uso do sistema, preferências e interesses e seus dados pessoais (idade, sexo e etc). Os sistemas de recomendação personalizadas devem construir o chamado "Perfil de Usuário", um modelo mantendo os dados sobre cada usuário do sistema. O modelo do Perfil do Usuário é implementado na forma de uma estrutura de dados, podendo ser incluindo na estrutura de um banco de dados.

O Perfil de Usuário mantém conceitos aproximados, ou seja, ele mantém o interesse do usuário sobre um conjunto de itens (produtos, músicas, vídeos e etc) em um momento particular (CAZELLA; NUNES, 2010). Cada termo em um Perfil de Usuário mantém uma informação diretamente solicitada ao usuário ou aprendida implicitamente com base no histórico de uso do sistema.

Em geral, um sistema de recomendação tem três operações principais: a captura do **Perfil do Usuário**, que mantém dados necessários à identificação das preferências/interesses do usuário; **Filtragem de Informação**, que seleciona os itens relevantes ao usuário utilizando

para isso os perfis dos usuários e dados de todos os itens; e **Ranqueamento dos Itens**, que gera uma lista ordenada dos itens que serão recomendados ao usuário.

Para a geração e manutenção do Perfil de Usuário se faz necessária a identificação de um determinado usuário. (CAZELLA; NUNES, 2010) evidencia duas formas de se identificar um usuário em sistemas Web:

- Identificação no servidor: se baseia na necessidade do usuário gerar manualmente um cadastro pessoal (informando dados como nome, login, senha, endereço e email). O usuário é identificado após ele ter realizado a sua autenticação no sistema (através do uso de login e senha);
- Identificação no cliente: se baseia de usos de mecanismos como cookies para detectar implicitamente o usuário do sistema. Este é um mecanismo mais simples e menos intrusivo, porém menos confiável que a identificação no servidor, pois parte do princípio que uma determinada máquina sempre irá representar o mesmo usuário.

A partir do momento em que o usuário é identificado, se faz necessária a captura de informações para a construção de seu perfil. Estas informações podem ser capturadas de duas formas (MIDDLETON; SHADBOLT; ROURE, 2004):

- Captura explícita: é quando o usuário deve informar explicitamente seus dados e preferências/interesses. Em geral, sistemas de recomendação usando tal forma de captura exigem que o usuário preencha um cadastro indicando sua idade, sexo, localização, formação, e realizar uma avaliação dos itens que ele acessa/usa. Esta forma de captura produz um modelo mais preciso do Perfil do Usuário, pois é obtida do próprio usuário. Um aspecto negativo desta forma de captura do perfil do usuário é que ela é muito intrusiva, exigindo um esforço grande do usuário, que muitas vezes ele não considera importante oferecer estes dados ao sistema. Devido a esta carga extra de trabalho, muitos usuários tendem a não informar tais dados, ou informar dados incorretos visando voltar rapidamente ao serviço que ele tenha interesse.
- Captura implícita: é quando o sistema de recomendação utiliza o histórico de interações do usuário com o sistema para tentar estimar suas preferências. Estas interações incluem as formas de

acesso aos itens (como visualização do item ou compra), forma de navegação no sistema e os critérios usados nas buscas realizadas. Essa forma tem a vantagem de ser não intrusiva, sem a necessidade de intervenção direta do usuário. Como aspecto negativo, ela tende a ser menos precisa que a construção explícita.

Algumas propostas realizam uma combinação das duas abordagens para formar o perfil do usuário. Estas propostas geralmente fazem a construção do perfil do usuário de maneira implícita inicialmente, possibilitando uma avaliação explícita para “remodelagem” do perfil após a recomendação.

3.2 ESTRATÉGIAS DE RECOMENDAÇÃO PERSONALIZADAS

Quando falamos sobre estratégias de recomendação, a forma de agrupar e apresentar as informações extraídas dos usuários é de suma importância. (CAZELLA; NUNES, 2010) define quatro principais estratégias de recomendação:

- **Reputação:** estas estratégias de recomendação se baseiam na ideia que os usuários avaliam explicitamente cada item após o acesso ou compra;
- **Recomendação por associação:** são as recomendações obtidas com base em técnicas capazes de identificar associações entre os itens avaliados pelos usuários a partir de consultas em bancos de dados. São formas complexas de recomendação, que exigem análises profundas dos hábitos do usuário para a identificação de padrões e recomendação de itens com base nestes padrões. Um exemplo é a recomendação do tipo “Clientes que compraram este produto também compraram estes produtos”, muito usual em sites de compras;
- **Associação por conteúdo:** estratégias de recomendação que se baseiam no conteúdo de determinado item, como tema, autor e compositor. Nesta estratégia de recomendação é necessário identificar associações em um escopo mais restrito. Também é muito usado em livrarias virtuais, com recomendações do tipo “Outros livros que abordam o mesmo tema tratado por este livro”;
- **Análise de sequência de ações:** nesta tipo de estratégia são observadas ações do usuário para tentar estimar o comportamento

de usuários através de históricos de atividade temporal, como em um histórico de compras de clientes. Informações encontradas nestas sequências temporais podem ser utilizadas na identificação de padrões de navegação e consumo, que em seguida podem servir aos sistemas de recomendação.

3.3 TÉCNICAS CLÁSSICAS DE RECOMENDAÇÃO

Os sistemas de recomendação podem adotar diferentes técnicas de Filtragem de Informação. As técnicas tradicionais são: Filtragem Baseada em Conteúdo (FBC), Filtragem Colaborativa (FC) e Filtragem Híbrida. Existem outras técnicas menos populares não por seu nível de importância, mas em geral por seu recente desenvolvimento.

3.3.1 Filtragem Baseada em Conteúdo

A FBC utiliza o Perfil de Usuário para identificar itens relevantes ao usuário, comparando o seu perfil aos valores dos metadados associados aos itens ou os próprios conteúdos dos itens. Em outras palavras, a FBC observa o comportamento do usuário e recomenda novos itens que tenham relação com o perfil do usuário (MIDDLETON; SHADBOLT; ROURE, 2004). (BALABANOVIC; SHOHAM, 1997) considera que um sistema de recomendação puramente construído na FBC é aquele em que as recomendações são feitas ao usuário baseando-se unicamente no perfil construído através da análise dos conteúdos dos itens que o usuário avaliou no passado.

De acordo com (BURKE, 2007), a FBC pode gerar recomendações com base em duas fontes de dados: características associadas aos itens (seja seu próprio conteúdo ou os metadados que o descrevem) e a avaliação dos itens pelo próprio usuário.

Uma das vantagens da FBC, segundo (MOONEY; ROY, 1999), diz respeito à privacidade de acesso a dados, pois a recomendação leva em consideração apenas o Perfil do usuário foco da recomendação. Isso é diferente da FC, onde a recomendação leva em consideração os perfis de outros usuários semelhantes e este fato pode evidenciar o que os usuários estão acessando. Outra vantagem da FBC, se comparada com a FC, é que itens que foram recentemente introduzidos no sistema são passíveis de recomendação. Isto ocorre porque a recomendação não requer a avaliação do novo item por outros usuários, sendo feito apenas

um cálculo de similaridade entre características do novo item (seus valores de metadados) e do perfil do usuário.

Uma das maiores deficiências da FBC é a chamada **Partida a Frio** (*Cold-Start*), que dificulta a recomendação nos estágios iniciais do uso do sistema. Este problema é bem conhecido e foi analisado em trabalhos como (BURKE, 2007) e (MIDDLETON; SHADBOLT; ROURE, 2004). O problema da partida a frio se deve ao fato de que os usuários recentes do sistema, ou usuários que utilizam muito pouco o sistema, tem um histórico de interação muito limitado, gerando assim um perfil do usuário capturado pouco preciso.

Outra desvantagem da FBC é o “fechamento dos itens a serem recomendados”, ou também chamado de “estreitamento da análise de conteúdos” (TORRES et al., 2004). Este problema se dá pela recomendação de itens similares em conteúdo com itens já avaliados ou acessados pelo usuário, impedindo a recomendação de itens que podem ter conteúdo diferente, porém são relacionados.

3.3.2 Filtragem Colaborativa

A FC é uma das técnicas amplamente utilizadas pelos sistemas de recomendação, inclusive sistemas utilizados por sites de comércio virtual (DANER, 2009). Na FC, os itens a serem recomendados a um usuário são determinados com base na identificação de grupos de usuários com perfis similares ((DANER, 2009), (ODONOVAN et al., 2008) e (HUANG; LI; CHEN, 2005)). Este grupo, chamado geralmente de **Vizinhos Próximos**, é determinado para cada usuário através do uso de funções que determinam a similaridade entre avaliações ou acessos a conteúdos realizados pelo usuário foco da recomendação e pelos outros usuários do sistema. As técnicas de recomendação colaborativa assumem que existe uma grande probabilidade de que um usuário irá gostar de conteúdos que seus vizinhos próximos gostam.

Algoritmos de FC podem ser divididos em duas categorias (BOUTEMEDJET; ZIOU, 2008):

- Algoritmos baseados em memória: o cálculo da similaridade dos perfis de usuários é feito comparando-os às avaliações feitas por outros usuários similares, através de algoritmos de comparação que verificam o grau de similaridade entre perfis de usuários;
- Algoritmos baseados em modelos: estes algoritmos constroem um modelo estatístico que estima as preferências dos usuários. Eles

podem usar redes Bayesianas, redes de dependência e diagnóstico de popularidade.

Segundo (CAZELLA; NUNES, 2010), a FC baseia-se em três passos:

1. Cálculo do peso de cada usuário em relação à similaridade ao usuário alvo (métrica de similaridade);
2. Seleção de um subconjunto de usuários com maiores similaridades (vizinhos próximos) para considerar na predição;
3. Normalização das avaliações e computar as predições ponderando as avaliações dos vizinhos com seus pesos.

No primeiro passo a definição da similaridade pode ser realizada através da aplicação de diversos coeficientes de similaridade como Cosseno e Pearson, entre outros (CAZELLA; NUNES, 2010).

As técnicas de FC possuem alguns problemas bem conhecidos e analisados por (CLAYPOOL, 1999) e (D.H. et al., 2012). Um deles é o fato desta técnica não classificar as preferências de um usuário por sua própria opinião (seu perfil), mas estimá-la através de usuários que tem opiniões parecidas. No caso, a preferência de um usuário nem sempre reflete a de outro.

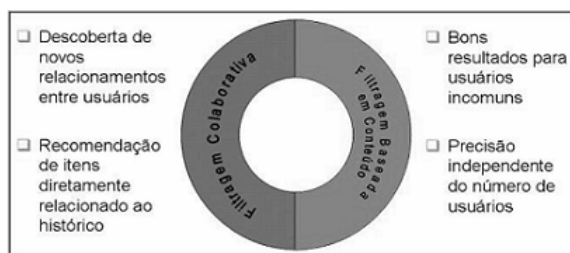
Outra deficiência conhecida da FC é problema de esparsidade, que se deve ao fato de que alguns itens acessados/adquiridos por um usuário podem ter tido pouco (ou talvez nenhum) acesso por outros usuários do sistema. Isso torna difícil a recomendação de itens similares a tais itens impopulares. Da mesma forma, itens pouco populares acabam sendo pouco (ou até nunca) recomendados. Consequentemente, conforme descrito por (BALABANOVIC; SHOHAM, 1997), se um novo item aparecer na base de dados, não há como ele ser recomendado enquanto nenhum usuário avaliá-lo.

A FC também sofre do problema da **Partida a Frio** (*Cold-Start*), que dificulta a recomendação no caso de usuários recentes no sistema, que possuem um perfil de usuário nulo ou muito esparsos. O problema da partida a frio se deve ao fato de que inicialmente existem poucos usuários que tenham definido suas preferências e, portanto, o sistema de recomendação tem dificuldade em encontrar usuários com perfis semelhantes devido à falta de informação que se encontra inicialmente.

3.3.3 Filtragem Híbrida

Apesar de muito difundidas e eficientes, as técnicas de FC e FBC também apresentam desvantagens uma em relação à outra. Por isso, para combinar as vantagens de ambas as técnicas, foram propostas algumas técnicas híbridas de recomendação, (TORRES et al., 2004), (VELLINO; ZEBER, 2007). A Figura 3 ilustra o conceito por trás desta ideia, ou seja, ela lista as principais vantagens, em unir a FBC com a FC.

Figura 3 – Recomendação híbrida.



Fonte: (CAZELLA; NUNES, 2010)

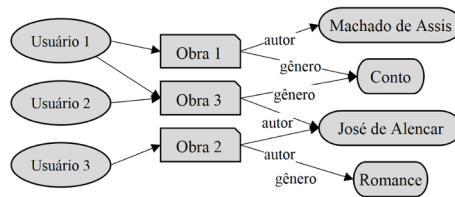
Estas técnicas híbridas geram a recomendação baseando-se tanto no perfil do usuário que busca a recomendação (FBC), como no perfil de outros usuários similares (FC). De acordo com (BURKE, 2007), sistemas híbridos tentam melhorar o desempenho geralmente para lidar com o problema da partida a frio. Por sua vez, (MOONEY; ROY, 1999) conclui que as FBC e FC são complementares e que, juntando ambas as técnicas (abordagem híbrida), pode-se conseguir resultados mais satisfatórios para a recomendação.

3.4 RECOMENDAÇÃO BASEADA EM GRAFOS

Vários trabalhos já demonstraram que sistemas de recomendação baseados em grafos produzem resultados promissores ((ANJORIN et al., 2012), (DEMOVIC et al., 2013) e (HUANG et al., 2014)). Na recomendação baseada em grafos o perfil dos usuários e/ou os descritores dos itens a recomendar são modelados na forma de grafos. De maneira geral, o objetivo da representação em grafos dos dados utilizados pelo sistema de recomendação é otimizar o processo de recomendação.

Em geral, os sistemas de recomendação baseados em grafos adotam técnicas de filtragem de informação conhecidas, como a FC, FBC ou FH. No caso da FC, pode-se utilizar um modelo de dados na forma de um grafo bipartido, com arestas conectando nodos usuários e itens (DEMOVIC et al., 2013). No caso da FBC aplicada à RDs, um grafo bipartido pode representar itens e valores de metadados como nodos e as arestas ligando itens a estes valores. Finalmente, uma FH baseada em grafos poderia adotar uma combinação dos três tipos de nodos (usuários, itens e metadados) em um grafo tripartido, como ilustrado na Figura 4. Alternativamente, o grafo pode representar as relações de similaridade entre usuários e entre itens.

Figura 4 – Modelo de dados representado em grafo



Fonte: Elaborada pelo autor

Em (ANJORIN et al., 2012), os autores propõem uma FH onde os nodos do grafo representam usuários e itens, e as arestas representam transações. Nodos e arestas possuem pesos e a recomendação é determinada através do algoritmo PageRank (BRIN; PAGE, 2012), que produz um ranqueamento dos nodos do grafo.

Existem também alguns sistemas de recomendação para vídeos baseados em grafos ((HUANG et al., 2014), (DEMOVIC et al., 2013)). Em (HUANG et al., 2014), os autores utilizam um grafo tripartido com os nodos usuário, item e consulta. Com base neste grafo é realizada uma FH implementada na forma de um algoritmo de propagação de mensagens geradas a partir dos nodos vídeos que se propagam nos nodos conectados, permitindo determinar o grau de interesse do usuário pelos vídeos. Já em (DEMOVIC et al., 2013), os nodos do grafo representam usuários, vídeos e seus metadados. Com base nesta representação, os autores propõem alguns algoritmos de recomendação. Um destes algoritmos é o *Energy Spreading Activation*, que atribui para cada nodo inicial um valor de energia constante E , e propaga esta energia para cada um dos N nodos filhos a uma taxa de E/N .

Maior parte dos trabalhos citados realizam análises teóricas so-

bre grafos já formados. Além disso, são propostas geralmente limitadas a um domínio, onde os nodos e arestas representam conceitos e propriedades predeterminados.

3.5 RECOMENDAÇÃO BASEADA EM ONTOLOGIAS

As tecnologias da Web Semântica também vêm sendo usadas por Sistemas de Recomendação. Uma ontologia é uma representação formalizada e não ambígua do conhecimento de um domínio, definindo conceitos, propriedades e relacionamentos (BORST, 1997). Com seu uso em sistemas de recomendação, é possível explorar o relacionamento entre conceitos e a semântica das propriedades. Desta forma, pode-se identificar interesses de usuários que não são diretamente observados. Por exemplo, um usuário acessando várias obras literárias de autores de um determinado estado pode indicar que este usuário tem interesse por obras deste estado. Este tipo de preferência muitas vezes não é capturado em sistemas de recomendações tradicionais, que observam apenas propriedades dos itens acessados, e não relações indiretas do item com outras entidades.

Em (SIEG; MOBASHER; BURKE, 2010), os autores propõem uma FC apoiada por perfis ontológicos de usuários. Para tal, é definida uma ontologia de domínio descrevendo conceitos relacionados aos itens a recomendar. O algoritmo *spreading activation* é utilizado para distribuir o grau de interesse associado a um conceito para os conceitos relacionados a este. Em seguida, a determinação dos vizinhos próximos na FC é realizada com base na similaridade das pontuações de interesse de cada usuário em cada um dos conceitos.

Similar a (SIEG; MOBASHER; BURKE, 2010), em (CANTADOR; CASTELLS; BELLOGÍN, 2011) os espaços de usuários e itens a recomendar são conectados por um espaço semântico. A partir desta representação, os autores propõem vários métodos de recomendação. Um destes métodos é a FH, que identifica comunidades de interesse com base na similaridade dos graus de interesse que cada usuário tem por cada um dos conceitos.

Tanto (SIEG; MOBASHER; BURKE, 2010) quanto (CANTADOR; CASTELLS; BELLOGÍN, 2011) não demonstram as vantagens do uso da ontologia em diferentes domínios, ou seja, não exploram a interoperabilidade semântica provida pelas ontologias. Neste sentido, é necessária a definição de uma ontologia de aplicação restrita a sistemas de recomendação. Uma iniciativa neste sentido foi realizada por (JACOBSON;

FERRIS, 2010), que propôs uma ontologia de recomendação especificando conceitos básicos e propriedades para descrever recomendações em alto nível para a Web Semântica. É importante observar que esta ontologia especifica o resultado da recomendação em si, não especificando conceitos que pudessem ser utilizados pelos sistemas de recomendação para apoiar a produção da recomendação em si.

3.6 INTEGRAÇÃO DE SISTEMAS DE RECOMENDAÇÃO EM RDS

Existem diferentes soluções para integrar sistemas de recomendação em RDs. Esta seção analisa três alternativas: frameworks de recomendação; *add-ons* para RDs; e *Web Services* de recomendação.

3.6.1 Frameworks de Recomendação

Frameworks de recomendação são soluções que facilitam o reuso de bibliotecas na implantação de sistemas de recomendação. Os três frameworks mais populares são *Apache Mahout* (OWEN et al., 2011), *LensKit* (EKSTRAND et al., 2011) e *MyMediaLite* (GANTNER et al., 2011).

Apache Mahout Apache Mahout é um projeto de fonte aberta pela Apache Software Foundation (ASF) com o objetivo primário de criar algoritmos de aprendizagem por máquina escaláveis que sejam livres para ser usados sob a licença Apache. O Mahout contém implementações para armazenamento em cluster, categorização, Filtragem Colaborativa e programação evolucionária. Além disso, quando prudente, ele usa a biblioteca Apache Hadoop para permitir que o Mahout escale de forma efetiva na nuvem. O framework roda sob JDK 1.6 ou versão superior.

LensKit fornece implementações cuidadosamente ajustadas dos principais algoritmos de filtragem colaborativa, APIs para casos comuns de uso de sistemas de recomendação e uma estrutura de avaliação para executar avaliações de algoritmos reproduzíveis offline. Seu objetivo principal é replicar e comparar resultados de pesquisas no campo referente a sistemas de recomendação.

MyMediaLite é uma biblioteca para sistemas de recomendação para Common Language Runtime (CLR, muitas vezes chamado .NET). Ele aborda os dois cenários mais comuns na filtragem colaborativa: Predição de classificação (em uma escala de 1 a 5 estrelas) e predição a partir de feedback positivo (cliques, curtidas ou ações de compra).

MyMediaLite é um software livre, que pode ser usado e distribuído sob os termos da Licença Pública Geral GNU (GPL). Escrito em *C#*, para a plataforma .NET, é executado em todas as arquiteturas suportadas pelo Mono: Linux, Windows, Mac OS X.

Esta solução permite o reuso de componentes de softwares escaláveis e eficientes para recomendação. De qualquer forma, a simples utilização destes frameworks torna a solução de recomendação acoplada à estrutura de dados dos RDs (como fonte de dados). Outro fator relevante é a necessidade da configuração e adaptação dos dados dos itens e usuários ao framework selecionado, bem como em todos os casos, os algoritmos abordados focam, quase que exclusivamente em filtragem colaborativa. Esta solução também não é recomendada para recomendação em RDs federados, por ser uma solução específica para um RD.

3.6.2 Ad-Ons de Recomendação para RDs

A Maior parte das soluções de RDs abertos são extensíveis, a partir do uso de *add-ons*. Portanto, uma forma de reduzir o esforço de implantação de serviços de recomendação em RDs é implementar este serviço na forma de um *add-on* de recomendação para RDs abertos.

Um dos poucos trabalhos neste sentido é o *add-on* para DSpace proposto em (ELLIOTT; RUTHERFORD; ERICKSON, 2008). Os autores apresentam Quambo, um add-on de recomendador para a plataforma DSpaceopen. Quando gera recomendações baseadas em conteúdo com base em um conjunto de exemplos selecionado pelo usuário, utilizando basicamente as palavras-chave do conteúdo selecionado. Esta alternativa é limitada a uma determinada solução de RD e também não suporta a recomendação em federações de RDs, tampouco filtragens colaborativas. Outro ponto a ser observado é o acoplamento da técnica a uma versão específica do Dspace.

3.6.3 Serviço de Recomendação via Web Services

Uma solução oferecendo um maior desacoplamento entre RDs e Sistemas de Recomendação é oferecer este último através de *Web Services* ((ANJORIN et al., 2012),(BEHAM et al., 2010) e (LEMOs et al., 2012)). Por exemplo, em (LEMOs et al., 2012) os autores propõem um sistema de recomendação de fotos baseado em *RestFul Web Service*, permitindo

que dispositivos móveis de baixa capacidade de processamento possam usar o serviço. Entretanto este trabalho visou um único domínio de aplicação (fotos) e os dados utilizados para recomendação são obtidos de bases de fotos específicas.

Em (ANJORIN et al., 2012), os autores propõem um sistema de recomendação de objetos de aprendizagem na forma de *Web Services* para possibilitar o seu uso por várias plataformas de ensino. Mas este serviço é exclusivo para recomendação de objetos de aprendizagem.

O sistema de recomendação APOSTDL (BEHAM et al., 2010) também é oferecido como um *Web Service*. Seu propósito é publicar recomendações de especialistas. Para tal, o serviço se baseia em uma ontologia de domínio que representa tarefas em um domínio de aprendizagem. O perfil de usuário é implicitamente capturado observando as tarefas realizadas pelo usuário.

Por fim, o sistema de recomendação Easyrec (EASYREC, 2017) é um *Web Service* de código aberto que fornece recomendações personalizadas usando *RESTful Web Service*. Trata-se de um sistema de propósito geral, para incorporação em sistemas web. A ferramenta provém recomendações baseadas visualizações, classificações e compra de conteúdos. As recomendações se baseiam na observação das ações dos usuários com determinados itens (outros usuários também visualizaram, itens classificados como bons por outros usuários, outros usuários também compraram e etc). Este *Web Service* fornece uma grande quantidade de ferramentas úteis, principalmente a sites do tipo e-commerce, porém abre mão da relação dos metadados de itens para manter a simplicidade da proposta. O Easyrec também requer informações diretamente associadas ao item acessado pelo usuário, como a descrição do item (nome utilizado pelo cliente para descrever o item) e a url de acesso do mesmo, expondo assim algumas informações que, dependendo do domínio da aplicação, podem ser sigilosas.

As soluções de sistemas de recomendação que se baseiam em *Web Services* oferecem um baixo acoplamento entre o sistema cliente do serviço de recomendação e o servidor oferecendo o serviço de recomendação. Além disso, elas oferecem formas de prover a recomendação em federações, pois os dados e usuários podem ser combinados pelo servidor.

As propostas encontradas na literatura apresentam duas características distintas: oferecem um sistema de recomendação mais completo, visando diferentes usuários, recomendações e metadados, porém focam em um domínio específico ou, na segunda situação, um sistema mais simples, visando a fácil implantação (com foco comercial), e dei-

xando de fora características possivelmente úteis para a inferência de conteúdos.

3.7 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais conceitos envolvendo sistemas de recomendação, desde os principais algoritmos bem como como aplicações e usabilidade. Foram apresentadas diversas técnicas pesquisadas e discutidas seus pontos fracos e fortes. Tendo em vista o objetivo deste trabalho que é propor uma técnica de recomendação de baixo acoplamento e independente de domínio voltada para repositórios digitais, este trabalho se destaca pois, diferente dos trabalhos, ele propõe um *Web Service* de recomendação independente de domínio, onde é possível especificar os conhecimentos do domínio relacionado aos itens a recomendar via a descrição de uma base de conhecimento ontológica.

4 UM WEB SERVICE DE RECOMENDAÇÃO PARA RDS

Os Repositórios Digitais (RDs) vem sendo cada vez mais utilizado por diversas organizações para disponibilização na Web de vários tipos de conteúdos digitais, incluindo artigos científicos, imagens, música, vídeos, produções literárias, e outros tipos de documentos. Como já citado, as coleções dos RDs tendem a crescer, principalmente em repositórios institucionais. A título de exemplo, o Repositório da Produção Científica e Intelectual da Unicamp (<http://www.repositorio.unicamp.br/> e da USP (<http://www.producao.usp.br/>) possuem atualmente mais de 130000 itens, o da UFSC (<http://repositorio.ufsc.br/>) conta com mais de 92000 itens. Apesar destes RDs possuírem ferramentas de busca e navegação nas coleções, seus usuários podem sofrer com frequência do problema de sobrecarga de informações.

Devido à tendência de aumento das coleções em RDs, a integração de sistemas de recomendações neste tipo de sistema é essencial para apoiar o usuário na localização de conteúdos relevantes. Como visto no capítulo anterior, as soluções abertas de RDs não dispõem atualmente de funcionalidades de recomendação personalizadas. Em geral, são oferecidas apenas indicação de submissões recentes e lista de itens populares.

Como também tratado pelo capítulo anterior, existem algumas soluções de integração de sistemas de recomendação em RDs, e a forma mais eficiente é via o uso de *Web Services* de recomendação. Neste sentido, e motivado pela carência de soluções de *Web Services* de recomendações facilmente adaptáveis ao domínio de conhecimento relacionado às coleções dos RDs, este trabalho propõe um *Web Service* de recomendação baseado em ontologia.

Desde já é importante observar que o *Web Service* de recomendação não é um Serviço Web Semântico. As ontologias são utilizadas aqui como forma aberta de especificar as informações relacionadas aos itens do repositório. Trata-se de uma forma de representação formalizada das informações sobre os usuários e conteúdos necessárias para a realização da filtragem de informação.

Este trabalho propõe uma Ontologia de Recomendação (RecOnt) que especifica conceitos e propriedades relacionadas à tarefa de recomendação. Trata-se de uma Ontologia de Tarefa (*Task Ontology*), que segundo (GUARINO, 1998) descreve um vocabulário relacionado a uma tarefa genérica, no caso, uma tarefa de recomendação de itens.

Outra característica relevante é que o *Web Service* de recomendação proposto é baseado em banco de dados orientado a grafos, que torna o uso de técnicas de recomendação orientadas a grafos seção (3.4 mais eficiente. Como consequência quase natural, este trabalho adota um banco de dados orientados a grafos, almejando garantir o desempenho do serviço.

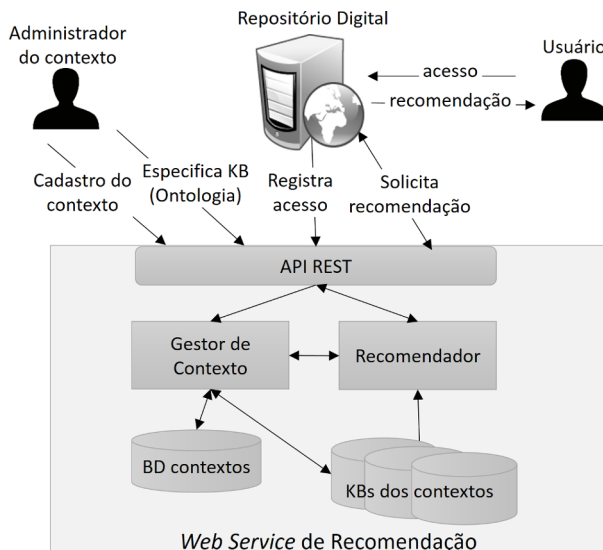
Este capítulo tem por objetivo detalhar o *Web Service* de recomendação proposto, iniciando pela apresentação de uma visão geral do serviço proposto.

4.1 VISÃO GERAL DO SERVIÇO PROPOSTO

A Figura 5 apresenta a arquitetura simplificada do *Web Service* de recomendação proposto. Nela são identificados os principais atores envolvidos, que são:

- *Administrador do Contexto*: é o usuário administrador do sistema cliente do serviço de recomendação. Ele tem a função de cadastrar o contexto na qual o serviço de recomendação será oferecido (*Cadastro do Contexto*) e de especificar a base de conhecimento (KB - *Knowledge Base*) inicial utilizada no processo de recomendação (*Especifica KB*). O contexto define as circunstâncias em que se dará a recomendação, em particular, a identificação do RD e o domínio de conhecimento relacionado com os itens disponibilizados pelo RD. A especificação da KB permite ao administrador especificar os conceitos usados na descrição de itens. Para tal, os administradores devem reusar, ou especificar uma ontologia de domínio, além de estender esta ontologia de domínio com a RecOnt;
- *Usuário*: representa os usuários acessando os conteúdos em um RD (*acesso*) e recebendo recomendações (*recomendação*). Estes usuários pertencem a uma determinada audiência de recomendação. Este trabalho utiliza o termo audiência para referenciar o conjunto de usuários foco de uma recomendação. Esta generalização é necessária, pois a audiência de um sistema de recomendação pode ser um determinado usuário (uma recomendação personalizada), um determinado grupo de usuários (habitando um determinado local geográfico, de determinada faixa-etária, etc.), ou ainda, uma recomendação única para todos usuários (no caso de uma recomendação por popularidade);

Figura 5 – *Web Service* de Recomendação Proposto



Fonte: Elaborada pelo autor

- *Repositório Digital*: é o lado cliente do serviço de recomendação, tendo como função registrar acessos a itens do RD realizados pela audiência da recomendação em um dado contexto (*Registra acesso*) e também solicitar recomendações a determinada audiência (*Solicita recomendação*);
- *Web Service de Recomendação*: é o serviço de recomendação oferecido a partir de uma API REST. Internamente, ele é formado pelo módulo *Gestor de Contexto*, que controla todos os aspectos relacionados ao contexto da recomendação, e pelo *Recomendador* que implementa os algoritmos de recomendação. As informações são mantidas em vários bancos de dados;
- *Gestor do Contexto*: responsável pela autenticação dos clientes e pelo processamento e armazenamento das informações de contexto. Ele inclui componentes responsáveis pela validação da ontologia de contexto e sua persistência na *BD contextos*. Este módulo também é responsável pelo mapeamento de ontologias em grafos, e pelo gerenciamento dos grafos mantidos nas *KBs dos contextos*.

- *Recomendador*: responsável pelo processamento dos algoritmos de recomendação solicitados pelo cliente para um determinado contexto. Dada a invocação do método de recomendação do *Web Service*, este módulo seleciona e executa o algoritmo de recomendação considerando a audiência objeto da recomendação. O resultado da execução do algoritmo é a lista de itens recomendados em ordem de relevância no formato JSON.
- *BD de contextos*: mantém os dados gerais dos contextos já cadastrados, incluindo dados de autenticação dos clientes e a identificação da KB de cada contexto.
- *KBs dos contextos*: mantém as bases de conhecimento dos contextos já cadastrados. Cada KB mantém, na forma de grafos, a ontologia de um dado contexto, os indivíduos representando as audiências, itens e seus atributos. Estes são os dados utilizados pelos algoritmos de recomendação.

Note que um contexto de recomendação considera um conjunto dos conteúdos digitais que são passíveis de serem recomendados (que são mantidos na KB). O administrador pode definir quantos contextos de recomendação forem necessários, para prover recomendação limitada a uma determinada coleção do RD, a um RD inteiro, ou então um contexto pode estar associado a uma federação de RDs. Neste último caso, a Figura 5 apresentaria um dos RDs da federação.

4.2 MODELAGEM CONCEITUAL DOS DADOS

No âmbito deste trabalho, uma KB de contexto de recomendação é especificada por uma ontologia de recomendação RecOnt, uma ontologia de domínio do contexto e pelos indivíduos categorizados pelos conceitos definidos nestas ontologias. Para a representação das ontologias de domínio e de recomendação foi adotada a linguagem OWL (*Web Ontology Language*) (GROUP, 2012).

A ontologia RecOnt trata-se de uma Ontologia de Tarefa (*Task Ontology*) (GUARINO, 1998) que descreve um vocabulário relacionado à tarefa de recomendação de itens.

A ontologia de domínio do contexto especifica conceitos e relacionamentos relacionados ao domínio de conhecimento associado aos conteúdos digitais de um contexto de recomendação. Por exemplo, no caso de obras literárias, a ontologia de domínio de contexto pode representar obras, autores, gêneros literários, dentre outros. Já um RD de

teses e dissertações, a ontologia de domínio pode representar autores, universidades, palavras-chaves, etc. Estas ontologias de domínio podem estar acompanhadas de indivíduos, que representam os conteúdos digitais e suas propriedades.

4.2.1 RecOnt: Ontologia de Recomendação

Para prover o serviço de recomendação, não basta a especificação do conhecimento relacionado ao contexto de recomendação (ontologia de domínio). É necessário identificar quais conceitos são itens que podem ser recomendados e quais conceitos devem ser levados em consideração para formação do perfil dos usuários. Para tal, este trabalho propõe uma ontologia de recomendação: a RecOnt.

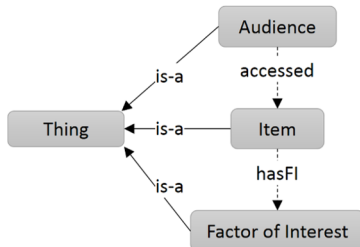
A Ontologia de recomendação RecOnt é uma ontologia de tarefa, ou seja, uma ontologia desenvolvida para uma tarefa específica, que é a de recomendação de itens. Ela descreve formalmente conceitos importantes na área de serviços de recomendação.

O Anexo A contém a representação OWL da ontologia RecOnt. A Figura 6 apresenta os principais conceitos desta ontologia, que são:

- *Audience*: representa o usuário ou o grupo de usuários foco da recomendação, sendo que uma audiência pode acessar itens. Este relacionamento é definido pela propriedade de objeto *accessed*, relacionando *Audience* com *Item*.
- *Item*: representa um item a ser recomendado, que é relacionado a pelo menos um *Factor de Interest* via a propriedade de objeto *hasFI*.
- *Factor of Interest*: representa alguma informação relacionada a *Item* que deve ser levada em consideração na recomendação (considerada relevante para determinação dos interesses/preferências do usuário).

Um aspecto importante a ser observado é que todos os conceitos da RecOnt (e qualquer outra ontologia OWL) são representados por URIs, tornando o sistema de recomendação desacoplado do RD ou federação de RDs. Itens a recomendar e audiências são identificados pelo sistema de recomendação como URIs. É específico de implementação do RD identificar o item ou usuário em si referenciado pela URI. Portanto, o sistema não captura nenhuma informação pessoal sobre o usuário e que tipo de conteúdo ele acessa.

Figura 6 – Ontologia RecOnt



Fonte: Elaborada pelo autor

Utilizando a RecOnt, os especialistas de domínio podem identificar conceitos nas ontologias de domínio a serem considerados pelo sistema de recomendação. Graças à RecOnt, a presente proposta é uma solução de propósito geral, que se adapta facilmente a qualquer domínio do conhecimento. Esta vantagem da presente proposta é ilustrada na sequência.

4.2.2 Ontologias de Domínio

Os RDs normalmente adotam esquemas de metadados para descrever os itens das coleções. O mais utilizado são os elementos Dublin Core (DMCI), apresentados na Seção 2.2.1. Estes itens são genéricos e podem ser utilizados para diferentes tipos de recursos digitais. Outros RDs adotam outros esquemas abertos, ou adotam extensões para possibilitar a definição de aspectos específicos dos itens dependendo do domínio de conhecimento relacionado ao item.

Os esquemas de metadados adotados por RDs podem ser representados por ontologias, como é o caso da proposta Dublin Core in OWL 2 (http://bloody-byte.net/rdf/dc_owl2d1/). Esta ontologia pode representar itens de maneira genérica, com os termos DMCI.

Outros RDs mantêm itens que são descritos por metadados muito relacionados ao domínio de conhecimento relacionados. A título de exemplo, considere a Biblioteca Digital de Literatura Brasileira (BDLB, apresentada na Seção 2.5), que adota metadados relacionados ao domínio da literatura. A partir destes metadados, pode-se definir uma ontologia de literatura simplificada, representando obras literárias, suas propriedades e outros conceitos relacionados.

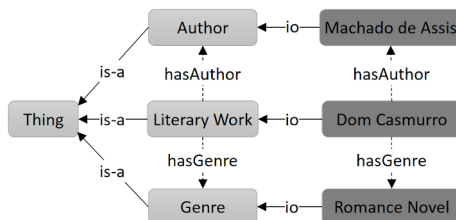
Para utilização da presente proposta de *Web Service* de reco-

mendação, o administrador deve inicialmente escolher, ou especificar, a ontologia de domínio do RD. Esta ontologia deve representar, na forma de conceitos e propriedades, todos os elementos do esquema de metadados adotada no RD que sejam relevantes para formação do perfil do usuário.

A título de exemplo, considere a ontologia de domínio na área de literatura presente no Anexo B e ilustrada na Figura 7. Esta ontologia simplificada descreve conceitos clássicos da área de literatura, como *Literary Work*, relacionada com metadados como *Genre* (como Romance, Conto e Novela) e *Author*. Esta ontologia pode ser adotada por RDs disponibilizando obras literárias, como é o caso da BDLB.

A Figura 7 especifica também três indivíduos a partir da ontologia de domínio defina: *Dom Casmurro*, uma instância de *Literary Work* (anotado por *io*); *Machado de Assis* da classe *Author*; e *Romance Novel* da classe *Genre*. Nesta figura é dado um nome aos indivíduos para facilitar o entendimento. Na realidade, eles podem ser referenciados por URIs, oferecendo maior privacidade em relação aos conteúdos acessados pelos usuários.

Figura 7 – Ontologia de Literatura



Fonte: Elaborada pelo autor

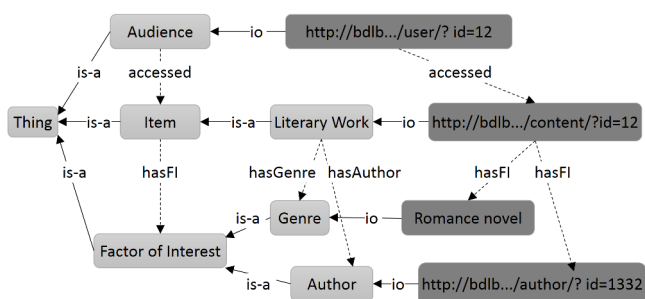
Como ilustrado na Figura 7, o administrador de contexto pode especificar dados sobre os itens a recomendar no momento de criação do contexto. Para possibilitar a construção e atualização do conhecimento, no momento da notificação de acessos a conteúdos, o RD deverá informar novos conhecimentos acerca dos itens acessados, ampliando assim dinamicamente, a base de conhecimento.

4.2.3 Ontologias de Contexto

Após especificada a ontologia de domínio, o administrador do contexto deve utilizar a RecOnt para identificar quais conceitos da on-

tologia de domínio devem ser considerados itens a recomendar e quais serão seus fatores de interesse. Por exemplo, o Anexo C apresenta uma ontologia de contexto de recomendação para o domínio de literatura. A Figura 8 ilustra esta ontologia, onde pode ser observado que o item selecionado para recomendação é *Literary Work* e os fatores de interesse a serem observados na recomendação são *Genre* e *Author*. A título de exemplo, a Figura 8 especifica também uma audiência que acessou uma obra, ambas referenciadas por URIs.

Figura 8 – Ontologia de Literatura estendida



Fonte: Elaborada pelo autor

Para criar a ontologia de Domínio e de Contexto, o administrador de contexto pode utilizar um editor de ontologia, como o Protégé (PROTÉGÉ, 2017).

4.3 INTERFACE DO SERVIÇO

A interface do serviço pode ser acessada por meio de *REST Web Services*. Os resultados das invocações dos serviços da plataforma são disponibilizados no formato JSON. Esta forma de representação foi adotada, em vez de uma representação semântica, por ser considerada mais simples de ser interpretada por humanos e máquinas.

O oferecimento do serviço via *Web Services* e estruturas JSON torna a solução independente de linguagem de programação/framework ou sistema operacional. O objetivo com isso é facilitar a incorporação do cliente do serviço de recomendação em um RD já existente.

O *Web Service* de recomendação proposto oferece métodos responsáveis pelo estabelecimento de contexto, manutenção da KB, bem como a solicitação de recomendações. Existem quatro métodos res-

ponsáveis por estas operações:

- **createContext**, que permite criar um novo contexto de recomendação;
 - **URL de acesso:** /createContext/apiKey/Context
 - **Método:** POST
 - **Parâmetros da url:**
 - * apiKey: Token que identifica o usuário com o serviço;
 - * Context: Parâmetro que identifica a qual contexto o serviço será criado;
 - **Mensagem de resposta:**
 - * 200 Context registered successfully;
 - **Mensagem de erro:**
 - * 400 Bad Request;
 - * 401 Unauthorized - Invalid user token;
 - * 422 Unprocessable Entity - Context already exists;
 - * 500 Internal Server Error
- **setKB**, para a criação e atualização da KB associada a um contexto. Os parâmetros deste método são os dados de autenticação do contexto e a KB especificada na linguagem OWL;
 - **URL de acesso:** /setKB/apiKey/Context
 - **Método:** POST
 - **Parâmetros da url:**
 - * apiKey: Token que identifica o usuário com o serviço;
 - * Context: Parâmetro que identifica a qual contexto o serviço será criado;
 - **Conteúdo da postagem:** Uma ontologia contendo a KB em OWL
 - **Mensagem de resposta:**
 - * 200 Ontology registered successfully;
 - **Mensagem de erro:**
 - * 400 Bad Request;
 - * 401 Unauthorized - Invalid user token;
 - * 422 Unprocessable Entity - Inconsistent ontology;

* 500 Internal Server Error

- **accessNotification**, que permite ao cliente registrar um dado acesso a um item por uma determinada audiência, enviando uma ontologia que registra o item que determinada audiência acessou, bem como seus fatores de interesse;

- **URL de acesso:** `accessNotification/apiKey/Context/actionTime`

- **Método:** POST

- **Parâmetros da url:**

- * apiKey: Token que identifica o usuário com o serviço;
- * Context: Parâmetro que identifica a qual contexto o serviço será criado;
- * actionTime: Momento em que a ação ocorreu;

- **Conteúdo da postagem:** Uma ontologia contendo o acesso de determinada audiência.

- **Mensagem de resposta:**

- * 200 Access registered successfully;

- **Mensagem de erro:**

- * 400 Bad Request;
- * 401 Unauthorized - Invalid user token;
- * 422 Unprocessable Entity - Inconsistent ontology/context;
- * 500 Internal Server Error

- **getRecommendation**, oferece o serviço de recomendação em si. Ele tem como parâmetros, além dos dados de autenticação no contexto, a URI da audiência da recomendação, e o algoritmo de recomendação a ser utilizado. O resultado é uma lista ordenada de itens recomendados.

- **URL de acesso:** `getRecommendation/apiKey/Context/user/nresult/rectype`

- **Método:** GET

- **Parâmetros da url:**

- * apiKey: Token que identifica o usuário com o serviço;
- * Context: Parâmetro que identifica a qual contexto o serviço será criado;

- * user: Audiência da recomendação;
- * nresult: quantidade de itens que serão retornados;
- * rectype: Tipo da recomendação, sendo 1 recomendação por popularidade, 2 recomendação baseada em conteúdo e 3 recomendação colaborativa;
- **Resposta do método:** Uma lista, no formato JSON com os itens solicitados.

Listagem 4.1 – Exemplo de resposta do método *getRecommendation*

```
{
  "audience": "38efcc71-46cd-667f99d427",
  "recommendationType": "2",
  "recommendation": {
    "entry": {
      "key": "1",
      "value": "4"
    }
  }
}
```

Na Listagem 4.1 é apresentado um exemplo de resposta contendo um item, o parâmetro *key* identifica a posição dele na lista, o parâmetro *value* identifica o item.

- **Mensagem de erro:**
 - * 400 Bad Request;
 - * 401 Unauthorized - Invalid user token;
 - * 422 Unprocessable Entity - Audience not found in context;
 - * 500 Internal Server Error

Uma alternativa ao uso dos métodos *createContext* e *setKB* é prover estas funcionalidades via páginas Web associadas ao serviço de recomendação. Elas permitiriam ao administrador do RD estabelecer com mais facilidade o contexto de recomendação.

4.4 REPRESENTAÇÃO NA FORMA DE GRAFOS

Um dos problemas clássicos dos sistemas de recomendação está relacionado à escalabilidade. Considerando este aspecto, nesta proposta optou-se em utilizar técnicas de recomendação baseados em grafos e o uso de banco de dados orientado a grafos. Esta escolha foi motivada pelos resultados promissores obtidos por trabalhos que adotaram o banco de dados orientado a grafos Neo4j em sistemas de recomendação ((BATRA; TYAGI, 2012) e (NEO4J, 2017)). De maneira geral,

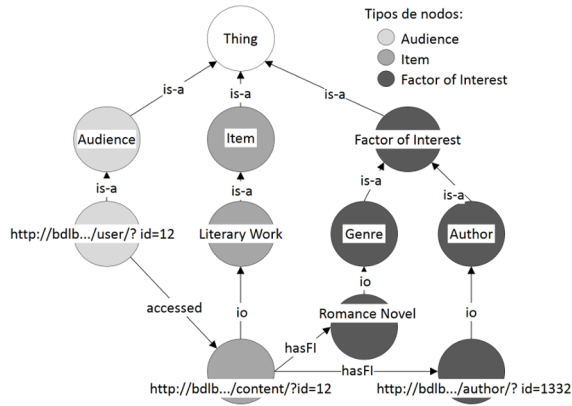
o objetivo da representação em grafos dos dados utilizados pelo sistema de recomendação é otimizar o processo de recomendação.

Apesar do uso de ontologias, não se cogitou neste trabalho o uso de tecnologias manipulando diretamente ontologias, como a OWL API (HORRIDGE; BECHHOFFER, 2011)). Isto devido ao fato que foram observados em projetos anteriores uma baixa escalabilidade destas tecnologias, o que tornaria inviável seu uso em RDs com grande volume de dados. A solução proposta utiliza apenas ontologias em suas interfaces, sendo as técnicas de recomendação implementadas baseadas em grafos. Para tanto, foi necessário a definição de uma técnica de mapeamento da ontologia OWL em grafos. Para este procedimento, a ontologia é serializada por meio da biblioteca OWL API, e em seguida é realizado o mapeamento para uma representação na forma de um grafo no banco de dados Neo4j. Para este mapeamento são realizados os seguintes passos:

1. O processo de mapeamento inicia com a representação da classe *Thing* da ontologia em um nodo não nomeado *Thing*. Este é o nodo raiz do grafo.
2. Todas as classes definidas na ontologia são representadas por nodos. As classes *Audience*, *Factor of Interest* e *Item* são representadas por nodos nomeados pelos respectivos nomes. É este fator que possibilita as técnicas de recomendação identificarem com facilidade o tipo de item, e como ele será processado. As demais classes são representadas por nodos não nomeados.
3. Para todos os nodos gerados, se a classe possuir superclasse, é definida uma aresta tipada com *is-a*.
4. Para cada indivíduo de uma classe, é gerado um nodo com uma aresta ligando-o ao nodo representando a classe. Esta relação é tipada com *io*.
5. Para cada indivíduo, são obtidas todas as propriedades de objetos e de dados para representá-las como propriedades ou relações dos nodos representando estes indivíduos.

A título ilustrativo, a Figura 9 apresenta o grafo resultante do mapeamento da Ontologia de Contexto de literatura (Figura 8). Este grafo representa o perfil dos usuários e os descritores dos itens a recomendar. Neste exemplo simplificado, o grafo representa apenas um único usuário e um único item.

Figura 9 – Grafo Resultante



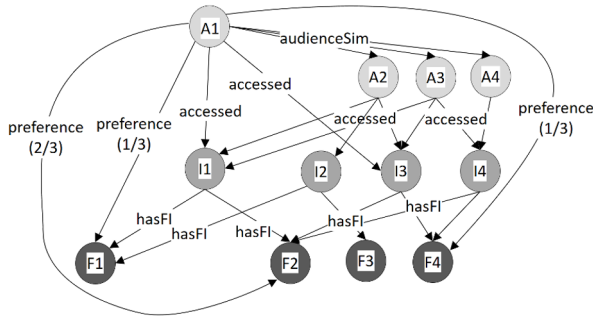
Fonte: Elaborada pelo autor

4.5 ALGORITMOS DE RECOMENDAÇÃO

Como visto na Seção 3.4, existem vários algoritmos de recomendação baseados em grafos. Note que este trabalho não tem por intenção propor novas técnicas de recomendação baseadas em grafos, mas prover uma solução de sistema de recomendação multidomínio fracamente acoplado a RDs. Para avaliar a factibilidade e escalabilidade da proposta, este trabalho adota e implementa algumas técnicas de recomendação: recomendação por popularidade, FBC e FC.

Para ilustrar as técnicas de recomendação baseada em grafo implementadas, considere o grafo da Figura 10, representando na forma de grafo o perfil de quatro audiências (A1, A2, A3 e A4) e quatro itens (I1, I2, I3 e I4). Por simplificação, considere que cada audiência de fato é um único usuário. Na Figura 10 foram omitidos os nodos representando as classes *Thing*, *Audience*, *Item* e *Factor of Interest*. Deve-se considerar que os nodos A1 a A4 possuem uma relação *io* com o nodo *Audience* e os nodos I1 a I4 tem relação *io* com o nodo *Item*. Os nodos FI1, FI2, FI3 e FI4 representa características dos itens (p.e., autor ou gênero do item) que possuem relação *hasFI* com estes. Finalmente, deve-se considerar que os nodos FI1 a FI4 têm relação *io* com o nodo *Factor of Interest*.

Figura 10 – Exemplo de modelagem do perfil dos usuários



Fonte: Elaborada pelo autor

4.5.1 Técnica de Recomendação por popularidade baseada em grafo

A recomendação baseada em popularidade é uma técnica não personalizada, que leva em conta apenas o número de acesso aos itens do RD. A lista de recomendação é formada por um número de itens mais populares, ordenados em ordem decrescente do número de acessos.

A implementação da técnica de recomendação por popularidade baseada no modelo proposto é simples. O número de acesso de cada item é obtido via a contabilização do número de arestas *accessed* do nodo com relação *io* com o nodo *Item*. Por exemplo, no caso da Figura 10, os itens *I1* e *I3* são os mais populares, com 3 acessos cada.

4.5.2 Técnica de Recomendação por FBC baseada em grafos

Como visto na Seção 3.4, nas técnicas FBC baseadas em grafos, o grafo geralmente representa itens e valores de metadados como nodos e as arestas ligando itens a estes valores. Este também é o caso do modelo proposto, em que os itens são representados por nodos *Item* (nodos com relação *io* com o nodo nomeado *Item*) (na Figura 10, os nodos *I1* a *I4*) e os diferentes valores de metadados dos itens já representados no grafo também são representados por nodos. Aqueles nodos representando valores de metadados que são considerados relevantes para a recomendação, também terão uma relação *hasFI* com o nodo *Item* descrito por este metadado (na Figura 10, os nodos *F1* a *F4*).

Conforme descrito na Seção 3.3.1, a FBC utiliza o Perfil de Usuário para identificar itens relevantes ao usuário, comparando o seu perfil aos valores dos metadados associados aos itens ou os próprios conteúdos dos itens. Para a comparação do perfil do usuário com os descritores dos itens é adotada uma certa função de similaridade.

Esta dissertação propõe uma FBC baseada em grafos baseada na técnica de recomendação proposta por (WILLRICH et al., 2006) e (CASAGRANDE; KOZIMA; WILLRICH, 2015), onde é introduzido o conceito de grau de preferência do usuário a um item com base na análise de frequência de ocorrência dos valores dos metadados nos itens acessados pelo usuário. Nesta técnica adaptada ao modelo proposto, este grau de preferência é representada por uma nova aresta nomeada *preference*, que liga todo nodo representando o usuário ou grupo de usuário com cada nodo representando o valor de metadado de um item já acessado pelo usuário. Em termos mais técnicos, a aresta *preference* liga todo nodo *Audience* (isto é, os nodos que tenham relação *io* com o nodo nomeado *Audience*) a todo nodo *Factor of Interest* (isto é, os nodos que tenham relação com o nodo nomeado *Factor of Interest*) que tenha relação *hasFI* com qualquer nodo *Item* (isto é, os nodos que tenham relação *io* com o nodo nomeado *Item*) que tenha relação *accessed* com o nodo *Audience*.

Uma aresta *preference* tem como atributo um valor numérico que corresponde ao grau de preferência. Este valor é calculado conforme a Equação 4.1.

$$pref(A_i, FI_j) = \frac{totalvalFI(A_i, FI_j)}{totalFI(A_i)} \quad (4.1)$$

Onde:

- A_i é um nodo representando uma audiência foco da recomendação (um usuário ou grupo de usuários), ou seja um nodo que tenha uma aresta *io* com o nodo nomeado *Audience*;
- FI_j é um nodo representando um valor de metadado associado com nodo item I_k acessado pelo usuário A_i , ou seja, um nodo que tenha aresta *accessed* com A_i e aresta *hasFI* com I_k ;
- $preference(A_i, FI_j)$ é o valor do atributo da aresta *preference* que liga o nodo A_i com o nodo FI_j ;
- $totalvalFI(A_i, FI_j)$ é uma função que retorna o número de arestas ligando os itens acessados (que tenham arestas *accessed*) pela audiência A_i ao *Factor of Interest* FI_j ;

- $totalFI(A_i)$ é uma função que retorna o número total de arestas ligando os itens acessados por A_i (que tenham arestas *accessed*) a nodos *Factor of Interest*. O objetivo desta divisão é a normalização dos valores;

A título de exemplo, considere o cenário ilustrado na Figura 10, onde são apresentadas as arestas *preference* apenas para a audiência $A1$ por motivo de simplificação. Esta audiência acessou dois itens, $I1$ e $I3$, um com os fatores de interesse $F1$ e $F2$, e o segundo com $F2$ e $F4$. Neste caso, o grau de preferência por estes fatores são $\frac{2}{3}$ para $F2$, $\frac{1}{3}$ para $F1$ e $F3$.

As arestas *preference*, e seus respectivos valores, são geradas/atualizados em modo *offline* a cada período de tempo (p.e., diariamente), de maneira a não comprometer o desempenho do sistema de recomendação. Este intervalo de tempo na atualização das arestas *preference* definirá o tempo de atualização dos perfis dos usuários. Caso seja realizado após cada acesso, a recomendação considera todo acesso de todos os usuários até o momento da recomendação. Mas, esta opção demanda mais recursos computacionais. Caso a atualização seja diária, as recomendações levarão em conta apenas acessos do dia anterior.

A FBC baseada é grafo proposta gera uma lista ordenada de itens mais similares aos itens já acessados pela audiência da recomendação (A_i). Para tal, são considerados os atributos das arestas *preference* de A_i com todos os nodos representando valores de metadados considerados fator de interesse. No caso, cada nodo *Item* não acessado por A_i (ou seja, que não tenha aresta *preference* ligando ele a A_i), terá um aresta *relevance* criada. A aresta *relevance* relaciona cada nodo *Audience* com todo nodo *Item* não acessado por esta audiência. As arestas *relevance* possuem um atributo numérico cujo valor é calculado conforme a Equação 4.2.

$$relevance(A_i, I_m) = \frac{somaPref(A_i, I_m)}{totalFI(A_i)} \quad (4.2)$$

Onde:

- A_i é um nodo representando uma audiência foco da recomendação (um usuário ou grupo de usuários), ou seja um nodo que tenha uma aresta *io* com o nodo nomeado *Audience*;
- FI_l é um nodo representando um valor de metadado associado com nodo item I_m que não tenha sido acessado pelo usuário A_i , ou seja um nodo que não tenha aresta *accessed* com A_i e tenha aresta *hasFI* com I_m ;

- $relevance(A_i, I_m)$ é o valor do atributo da aresta *relevance* que liga o nodo A_i com o nodo I_m ;
- $somaPref(A_i, I_m)$ é uma função que retorna a soma dos atributos das arestas *preference* todos os nodos representando valores de metadados de I_m (nodos com aresta *hasFI* com I_m) com o nodo audiência A_i ;
- $totalFI(A_i)$ é uma função que retorna o número total de arestas ligando os itens não acessados por A_i (que não tenham arestas *accessed*) a nodos *Factor of Interest*. O objetivo desta divisão é a normalização dos valores;
- *grau de relevância calculado*. Este grau é determinado somando-se os graus de preferência das arestas *preference* ligando os fatores de interesse do item à audiência A_i . Por exemplo, no caso da Figura 10, o item $I2$ terá um grau de relevância de $\frac{2}{3}$, e item $I2$ terá $\frac{1}{3}$.

Retomando o exemplo da Figura 10, caso o usuário foco da recomendação seja $A1$, na FBC proposta, a lista de recomendação será, na ordem de relevância, $I4$ e $I2$. Sendo que $I4$ terá uma relevância calculada de $\frac{\frac{1}{3} + \frac{2}{3}}{4}$ e $I2$ terá uma relevância de $\frac{\frac{1}{3} + 0}{4}$.

4.5.3 Técnica de Recomendação por FC baseada em grafo

Como visto na Seção 3.3.2, na FC, os itens a serem recomendados a um usuário são determinados com base na identificação de grupos de usuários com perfis similares, os Vizinhos Próximos. Para tal, é utilizada uma função de similaridade entre os perfis de usuários.

Na técnica de FC baseada em grafo proposta, é definida uma nova aresta, chamada *audienceSim* ligando cada audiência com cada outra audiência. Arestas deste tipo têm como propriedade um valor que define o grau de similaridade entre os dois usuários. Para determinação deste valor foi adotada a técnica proposta por (WILLRICH et al., 2006) e (CASAGRANDE; KOZIMA; WILLRICH, 2015).

Uma aresta *audienceSim* tem como atributo um valor numérico resultante do cálculo da função de similaridade. A função de similaridade adotada foi a fórmula do cosseno (também adotada por (CASAGRANDE; KOZIMA; WILLRICH, 2015)). No caso, o valor deste atributo é calculado conforme a Equação 4.3.

$$audienceSim(A_i, A_n) = \frac{\sum_{p=1}^P pref(A_i, FI_p) * pref(A_n, FI_p)}{\sqrt{\sum_{p=1}^P pref(A_i, FI_p)^2 * \sum_{p=1}^P pref(A_n, FI_p)^2}} \quad (4.3)$$

Onde:

- A_i e A_n são nodo representando duas audiências, ou seja dois nodos que tenham aresta *io* com o nodo nomeado *Audience*; N é o número de nodos representando valores de metadados dos itens acessados por A_i , ou seja, o número total de arestas *preference* de A_i ;
- $preference(A_i, FI_p)$ é o valor do atributo da aresta *preference* que liga o nodo A_n com o nodo FI_p ;
- $preference(A_n, FI_p)$ é o valor do atributo da aresta *preference* que liga o nodo A_p com o nodo FI_p ;
- $audienceSim(A_i, A_n)$ é uma função que retorna a similaridade entre o perfil da audiência A_i com a audiência A_n ;

A título de exemplo, considere novamente o cenário ilustrado na Figura 10, onde são apresentadas as arestas *audience* apenas para a audiência $A1$ com as outras audiências por motivo de simplificação. Os valores dos atributos das arestas *audienceSim* para os usuários $A2$, $A3$ e $A4$, são respectivamente, 0,37, 0,48 e 0,15.

A exemplo das arestas *preference* e *relevance*, as arestas *audienceSim*, e seus respectivos valores, são geradas/atualizados em modo *offline* a cada período de tempo.

Neste caso, os 'vizinhos próximos' são os K nodos *Audience* com maiores valores da aresta *AudienceSim* ligando estes à A_i . Em seguida, são tomados todos os nodos *Item* acessados pelos 'vizinhos próximos' e não acessados por A_i . Finalmente, estes nodos são ordenados tomando como base o grau de similaridade dos valores das arestas *AudienceSim* e popularidade do item dentre os 'vizinhos próximos'.

4.6 MUDANÇAS NECESSÁRIAS NO RD

Para fazer uso do *Web Service* de recomendação proposto, é necessário adaptar o código do RD para registrar os acessos realizados pe-

los usuários (chamadas a *accessNotification*) e solicitar recomendações a determinada audiência (chamadas à *getRecommendation*).

Para notificar os acessos aos conteúdos digitais pelos usuários, é necessário identificar a parte do código do RD que trata esta solicitação para que seja incluída uma chamada a *accessNotification*. Neste momento é necessário identificar os parâmetros deste método: URI identificando o conteúdo no RD; a identificação da audiência que pertence o usuário (um método que classifica o usuário em uma audiência); e as propriedades e indivíduos relacionados ao conteúdo acessado. Esta operação é ilustrada no próximo capítulo.

Na apresentação da recomendação, a lista de recomendação retornada pelo método *getRecommendation* (uma estrutura JSON) pode ser apresentada na forma de uma lista ordenada de itens recomendados, informando os títulos dos itens na forma de um link dando acesso ao conteúdo recomendado.

5 IMPLEMENTAÇÕES E AVALIAÇÕES

A solução proposta de um *Web Service* de Recomendação para RDs tem por principais vantagens seu baixo acoplamento com os RDs fazendo uso do serviço e também a independência de aspectos relacionados ao domínio de conhecimento relacionado com os itens dos RDs. Outra característica diferenciada da proposta é o uso de um modelo baseado em grafos para representação dos perfis de usuários e dados sobre os itens, que é persistido em um banco de dados orientado a grafos. Este último visa a otimização do tempo de processamento de algoritmos de recomendação.

Este capítulo apresenta a implementação de um protótipo prova-de-conceito do *Web Service* proposto, que é utilizado em dois estudos de caso. O objetivo aqui é demonstrar e avaliar as vantagens da solução proposta em termos de baixo acoplamento, independência de domínio e tempos de resposta.

Visando evidenciar o aspecto multidomínio da solução proposta, os dois estudos de caso são de domínios diferentes, com diferentes soluções de RDs. No primeiro estudo de caso, foi utilizada a Biblioteca Digital de Literatura Brasileira (BDLB), uma implementação *ad hoc* específica para o domínio de Literatura, onde os tipos de itens a recomendar são obras literárias. No segundo estudo de caso, foi utilizado a solução aberta de RD DSpace, que é uma implementação genérica de RD independente de domínio. Neste segundo caso, foi considerado que os itens a recomendar são trabalhos científicos (incluindo teses e dissertações). Em ambos os casos, a audiência da recomendação foi definida como tendo um único usuário, ou seja, recomendação personalizada.

Este trabalho apresenta também medições dos tempos de resposta do *Web Service* proposto, visando avaliar a escalabilidade do protótipo implementado. Note que o objetivo deste experimento não foi avaliar a qualidade das recomendações geradas, mas avaliar a factibilidade do *Web Service* de recomendação adaptável ao domínio e sua escalabilidade. Novos algoritmos de recomendação baseados em grafos, possivelmente mais eficientes podem ser facilmente desenvolvidos com a solução proposta.

5.1 PROTÓTIPO DO SERVIÇO DE RECOMENDAÇÃO

Para estes estudos de caso, foi desenvolvido um protótipo do *Web Service* de recomendação proposto. Trata-se de um protótipo prova-de-conceito, ou seja, ele implementa as principais funcionalidades do sistema proposto. O protótipo desenvolvido suporta os três algoritmos de recomendação apresentados na Seção 4.5. Estão fora do escopo desta implementação o tratamento de possíveis erros nas requisições do *Web Service*.

Para este protótipo, foram utilizadas as seguintes tecnologias:

- Linguagem Java para o desenvolvimento das funcionalidades do serviço, sendo que a máquina virtual adotada foi a *SE Runtime Environment* versão 1.9.0;
- Apache Tomcat versão 7, um implementação aberta utilizado como servidor de aplicações;
- Banco de Dados *Neo4j Community Edition* versão 2.3.7, utilizado para a persistência do modelo orientado a grafos;
- OWL-API versão 4.1.0 utiliza na verificação da consistência das ontologias e no mapeamento de ontologias em grafos.

Para mostrar o trabalho de desenvolvimento, foram desenvolvidas 781 linhas de código java (não foram contabilizadas linhas em branco, comentários e bibliotecas utilizadas).

5.2 INTEGRAÇÃO DO SERVIÇO DE RECOMENDAÇÃO NA BDLB

Como apresentado na Seção 2.5, a Biblioteca Digital de Literatura Brasileira (BDLB) está disponível em <http://www.literaturabrasileira.ufsc.br/> e oferece aos seus usuários uma vasta coleção de obras literárias e dados sobre escritores na sua maior parte brasileiros. Este RD oferece uma série de funcionalidades, que inclui o cadastramento de usuários para o uso de serviços como publicação de comentários e criação de anotações digitais nas obras disponibilizadas em código HTML. Além disso, este RDs disponibiliza originalmente um serviço de recomendação, implementado no contexto do trabalho de (CASAGRANDE; KOZIMA; WILLRICH, 2015). Este último é provido por uma implementação própria, fortemente integrada aos demais componentes da BDLB.

5.2.1 Ontologia de Domínio e Ontologia de Contexto

A primeira etapa para uso do *Web Service* de recomendação proposto é a definição da ontologia de domínio e de contexto de recomendação. Este estudo de caso, estas ontologias foram as mesmas utilizadas como exemplo na Seção 4.2, apresentadas nos Anexos B e C e nas Figuras 7 e 8.

5.2.2 Modificação no RD BDLB

Para prover seus serviços personalizados, a BDLB permite aos seus usuários realizarem seus cadastramentos. Atualmente, a BDLB conta com 1158 usuários cadastrados. É a partir do login do usuário que o sistema identifica o usuário para o oferecimento dos serviços de comentários, anotações digitais e recomendação.

No contexto desta dissertação, a BDLB foi alterada para atuar como um cliente do *Web Service* de recomendação proposto. A implementação da BDLB segue o padrão de arquitetônico MVC (**Model-View-Control**). O componente modelo (*Model*) trata dos dados da aplicação, regras de negócios, lógica e funções. Uma visão (*View*) são os componentes responsáveis pela exibição das interfaces com o usuários. O controlador (*Controller*) realiza a mediação da entrada, convertendo-a em comandos para o modelo ou visão.

Neste estudo de caso, como a BDLB já oferecia um serviço de recomendação, não foram necessárias alterações nas visões e modelo do sistema. Foi necessária apenas a modificação no controle relacionado ao acompanhamento dos acessos a itens por parte do usuário e relacionado à geração da lista de recomendação. Foram modificados dois métodos do controle de recomendação:

- *registraAcesso(\$idUserario, \$idItem)*: que é chamado quando o usuário acessa um item do repositório (obra literária). Como parâmetro, são passados o identificador do usuário na BDLB (*\$idUserario*) e o identificador do item acessado (*\$idItem*). Este método então acessa os valores de metadados da obra literária que são considerados fator de interesse, que são o(s) autor(es) e gênero(s) literário(s) da obra e os representa semanticamente usando a ontologia de contexto. Finalmente ele invoca o serviço **accessNotification** do *Web Service* de recomendação. A Listagem 5.1 apresenta um exemplo de invocação deste serviço.

- *obterRecomendacao(\$idUserario, \$tipo, \$limit)*: método utilizado para invocar o serviço *getRecommendation* do *Web Service* de recomendação. O parâmetro são o identificador do usuário que receberá a convocação (*\$idUserario*), o tipo de recomendação (*\$tipo*) e a quantidade máxima de itens a recomendar (*\$limit*). A lista de recomendação retornada pelo *Web Service* é expressa em uma estrutura JSON, sendo que os itens são identificados pelos identificadores dos itens. O metodo *obterRecomendacao* decodifica esta estrutura JSON e gera um array onde cada elemento possui, dentre outros, o identificador da obra, seu título e autores. A criação desta array é necessária para que o componente de visão apresente adequadamente a lista na interface Web. A Listagem 5.2 apresenta um exemplo de invocação do serviço *getRecommendation* e seu resultado.

Listagem 5.1 – Exemplo de invocação do método *accessNotification*

```
POST /WebServiceProject/wsTeste/accessNotification/litdigital/
repositorio3/1486378847 HTTP/1.1

..
Content-length: 2415

<?xml version="1.0"?> <!DOCTYPE Ontology [ <!ENTITY xsd "http://www.
w3.org/2001/XMLSchema#" > <!ENTITY xml "http://www.w3.org/XML
/1998/namespace" > <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-
schema#" > <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax
-ns#" > ]> <Ontology xmlns="http://www.w3.org/2002/07/owl#"
xml:base="http://literaturabrasileira.ufsc.br/access" xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:xml="http://
www.w3.org/XML/1998/namespace" xmlns:xsd="http://www.w3.org
/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#" ontologyIRI="http://literaturabrasileira.ufsc.br/access
"> <Prefix name="" IRI="http://literaturabrasileira.ufsc.br/
access#"/> <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl
#"/> <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"/> <Prefix name="xsd" IRI="http://www.w3.org/2001/
/XMLSchema#"/> <Prefix name="rdfs" IRI="http://www.w3.org
/2000/01/rdf-schema#"/> <Prefix name="RecOnt" IRI="http://biblio
.inf.ufsc.br/~anderson/RecOnt.owl#"/> <Prefix name="
LiteratureContext" IRI="http://biblio.inf.ufsc.br/~anderson/
LiteratureContext.owl#"/> <Import>http://biblio.inf.ufsc.br/~
anderson/LiteratureContext.owl</Import> <Declaration> <
NamedIndividual IRI="#U1"/> </Declaration> <Declaration> <
NamedIndividual IRI="#A1"/> </Declaration> <Declaration> <
NamedIndividual IRI="#G4"/> </Declaration> <Declaration> <
NamedIndividual IRI="#2"/> </Declaration> <ClassAssertion> <
Class abbreviatedIRI="RecOnt:Audience"/> <NamedIndividual IRI="#
```

```

U1"/> </ClassAssertion> <ClassAssertion> <Class abbreviatedIRI="
LiteratureContext:Author"/> <NamedIndividual IRI="#A1"/> </
ClassAssertion> <ClassAssertion> <Class abbreviatedIRI="
LiteratureContext:Genre"/> <NamedIndividual IRI="#G4"/> </
ClassAssertion> <ClassAssertion> <Class abbreviatedIRI="
LiteratureContext:Literary_work"/> <NamedIndividual IRI="#2"/> <
/ClassAssertion> <ObjectPropertyAssertion> <ObjectProperty
abbreviatedIRI="RecOnt:accessed"/> <NamedIndividual IRI="#U1"/>
<NamedIndividual IRI="#2"/> </ObjectPropertyAssertion> <
ObjectPropertyAssertion> <ObjectProperty abbreviatedIRI="
LiteratureContext:hasAuthor"/> <NamedIndividual IRI="#2"/> <
NamedIndividual IRI="#A1"/> </ObjectPropertyAssertion> <
ObjectPropertyAssertion> <ObjectProperty abbreviatedIRI="
LiteratureContext:hasGenre"/> <NamedIndividual IRI="#2"/> <
NamedIndividual IRI="#G4"/> </ObjectPropertyAssertion> </
Ontology>

```

Listagem 5.2 – Exemplo de invocação do método getRecommendation

```

GET /WebServiceProject/wsTeste/rec/litdigital/U1/5/1 HTTP/1.1
Host: biblio.inf.ufsc.br:8080
Accept: */*

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/json
Transfer-Encoding: chunked
Date: Mon, 06 Feb 2017 11:58:58 GMT

83
{"audience": "U1", "recommendationType": "1", "recommendation": {"entry":
  [{"key": "1", "value": "Literary_work"}, {"key": "2", "value": "2"}]}}
0

```

A Figura 11 apresenta a interface Web da BDLB contendo a lista de obras recomendadas ao usuário. No canto superior esquerdo aparece o usuário autenticado e na parte inferior central a recomendação gerada a partir do perfil deste usuário.

O impacto das mudanças necessárias na BDLB para uso do *Web Service* de recomendação proposto foram muito reduzidos. Isto porque na realidade a BDLB já possuía outro serviço de recomendação. Foram necessárias basicamente a modificação de dois métodos PHP já existentes na implementação. Foram necessárias apenas a implementação de 88 linhas de código.

Figura 11 – Interface de Usuário da BDLB com a recomendação



Fonte: Elaborada pelo autor

5.3 INTEGRAÇÃO DO SERVIÇO DE RECOMENDAÇÃO NO DSPACE

Neste estudo de caso foi adotado como cliente do serviço de recomendação a solução aberta de RD DSpace. Mais especificamente, foi considerada a customização do DSpace do Repositório Institucional da UFSC. Neste trabalho, a coleção considerada pelo serviço de recomendação foi a de Teses e Dissertações da UFSC.

5.3.1 Ontologia de Domínio e Ontologia de Contexto

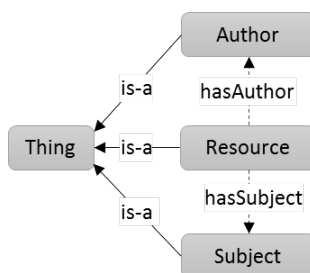
Para o uso do serviço de recomendação proposto, a primeira etapa é a definição da ontologia de domínio de conhecimento relacionado aos itens a recomendar. O DSpace segue o esquema de metadados DMCI (visto na Seção 2.2.1), que é suficientemente genérico para ser

usado para indexar diferentes tipos de recursos digitais. Neste estudo de caso, foi considerado como itens de recomendação as teses e dissertações do Repositório Institucional da UFSC.

Inicialmente foi realizado um estudo para identificar os metadados mais relevantes para serem considerados para recomendar teses e dissertações. Neste estudo foram considerados fator de interesse os metadados *Author* (corresponde a *dc:Creator*) e *Subject* (corresponde a *dc:Subject*). A partir disso, foi definida a ontologia de domínio DCRresource, apresentada no Anexo D e ilustrada na Figura 12. Os conceitos desta ontologia são:

- Resource: Representa um recurso no DSpace, sendo um recurso digital como um artigo, tese, dissertação, etc. Um Resource possui um ou mais autores (*hasAuthor*) e uma ou mais palavras-chave (*hasSubject*);
- Author: Representa um autor do recurso;
- Subject: Representa uma palavra-chave associada ao recurso.

Figura 12 – Ontologia de Domínio do Caso de Uso DSpace

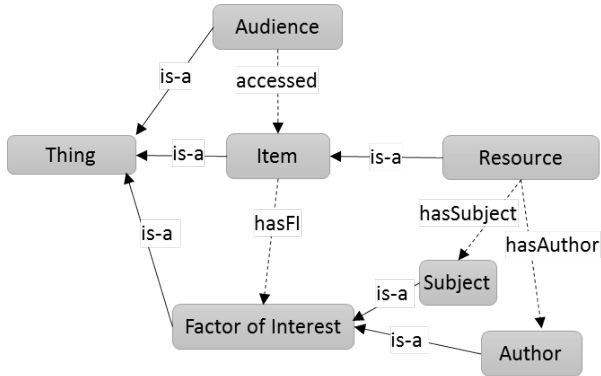


Fonte: Elaborada pelo autor

A partir da ontologia de domínio DCRresource, foi especificada ontologia de contexto, chamada DCContexto. Esta ontologia é apresentada no Anexo D e ilustrada na Figura 13. Ela define que o item a recomendar é da classe *Resource*, e que os fatores de interesse são *Author* e *Subject*.

A ontologia de domínio define conceitos gerais sobre o domínio dos recursos disponibilizados no repositório DSpace. Para este trabalho, adotou-se uma ontologia de domínio genérica, considerando conceitos genéricos existentes em repositório DSpace, chamada DCRresource.

Figura 13 – Ontologia de Contexto do Caso de Uso DSpace



Fonte: Elaborada pelo autor

5.3.2 Modificações no RD DSpace

Como visto na Seção 2.4.1.6, existem duas opções para a implementação das interfaces Web do DSpace: JSPUI (Java Server Pages User Interface) e XMLUI (eXtented Mark Language User Interface). Neste trabalho foi adotada o XMLUI, pois ela apresenta vantagens como a possibilidade de customização, oferecendo ao usuário a escolha de adotar características distintas para comunidades, coleções e itens. Outras inovações são a separação da camada de negócio da interface, internacionalização e localização de conteúdos e compatibilidade com a então atual interface baseada em JSP (WOLF; MONTEIRO; VALMORBIDA, 2013).

É importante destacar que este trabalho de modificação do DSpace foi desenvolvido por (ENGELKE; WILLRICH, 2016), no contexto do seu Trabalho de Conclusão de Curso da UFSC. Ele visou incorporar a funcionalidade de Sistema de Recomendação proposta nesta dissertação no RD DSpace.

Para incorporar as funcionalidades do Sistema de Recomendação proposto no DSpace foi necessário o uso de sua API e obedecer ao padrão de projeto adotado pelo DSpace. Portanto, foi necessário o desenvolvimento de um componente *Aspects* com a funcionalidade de recomendação de itens para o DSpace. O sistema implementado utiliza-se de duas classes Java inseridas no aspecto *viewArtifacts* localizado dentro do módulo *dspace-xmlui* (ENGELKE; WILLRICH, 2016):

- AccessRegister : Responsável pela coleta dos metadados do recurso, construção da ontologia e envio das informações via HTTP;
- RecOnt: Encarrega-se da solicitação dos dados no formato JSON a partir do servidor de recomendação e trata-os para a construção de recursos dentro do DSpace.

Existem dois cenários de apresentação da recomendação para os usuários do repositório. O primeiro cenário é quando o usuário do RD não está autenticado no sistema (usuário anônimo). Este tipo de recomendação é utilizado, pois o sistema não pode identificar o usuário, e, portanto, não é possível usar recomendações personalizáveis. Neste caso é utilizada a recomendação por popularidade. Na recomendação para um usuário anônimo, o sistema deve realizar as seguintes ações (ENGELKE; WILLRICH, 2016):

- Solicitar a recomendação por Popularidade ao *Web Service* de recomendação. Esta chamada utiliza como identificador do usuário o termo *anonymous*;
- Receber a lista de recursos recomendados na forma de uma estrutura JSON. Cada item será identificado pelo seu *handle*;
- Geração dinâmica da página HTML contendo a lista de recomendação na forma de uma div. Para tal, utilizando o *handle* de cada item a recomendar, o sistema deverá recuperar o título e os autores do recurso.

A Figura 14 apresenta a interface do DSpace, onde pode-se ver a apresentação da recomendação baseada em popularidade.

No caso do usuário se autenticar no DSpace, o sistema utiliza a recomendação FBC. Neste caso, o processo de recomendação deve realizar as seguintes ações (ENGELKE; WILLRICH, 2016):

- Solicitar a recomendação por FBC utilizando o *Web Service* de recomendação. Conforme previsto na API, esta chamada deverá conter o identificador do usuário e o identificador do tipo de recomendação solicitada;
- Solicitar a recomendação por Popularidade ao *Web Service* de recomendação. Esta chamada utiliza como identificador do usuário o termo *anonymous*;
- Receber a lista de recursos recomendados na forma de uma estrutura JSON. Cada item será identificado pelo seu *handle*;

Figura 14 – Apresentação da recomendação no DSpace.



Fonte: (ENGELKE; WILLRICH, 2016)

- Geração dinâmica da página HTML contendo a lista de recomendação na forma de uma div. Para tal, utilizando o *handle* de cada item a recomendar, o sistema deverá recuperar o título e os autores do recurso.

5.4 AVALIAÇÕES DO SERVIÇO DE RECOMENDAÇÃO

Como já citado, o presente trabalho não tem por objetivo definir novas técnicas de recomendação, mas sim definir uma solução de serviço de recomendação para RDs. Portanto, esta seção não tem como propósito avaliar qualidade dos algoritmos de recomendação. A meta é avaliar empiricamente a escalabilidade do sistema proposto.

Nesta seção são apresentadas duas avaliações realizadas. A primeira, visando avaliar a escalabilidade das diferentes técnicas de recomendação baseadas em grafos apresentadas na Seção 4.5. Em seguida, são apresentados os tempos de resposta da recomendação na BDLB (Seção 5.2).

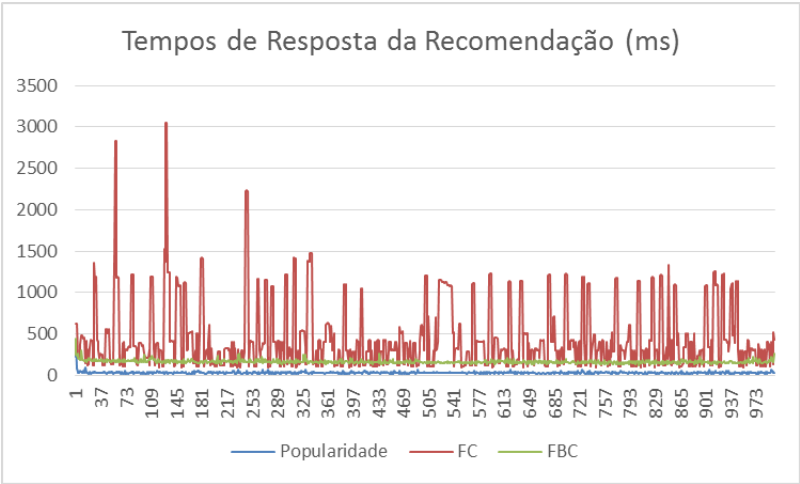
5.4.1 Avaliação das Técnicas de Recomendação

Neste primeiro experimento, o protótipo do *Web Service* de recomendação foi instalado em um computador com processador Intel i5 750 (8M Cache, 2.66 GHz) com 8 GB de memória. Em seguida, foi simulado um cenário hipotético onde 10.000 usuários acessam 10 itens aleatórios em um RD fictício contendo ao todo 1000 itens ao todo.

Este teste avalia o desempenho das técnicas de recomendação a partir do modelo de dados proposto, sem levar em consideração o *Web Service*, ou seja, os tempos de resposta aqui registrados são referentes ao período após a solicitação da recomendação e antes do envio da resposta ao cliente, de forma a avaliar o desempenho dos algoritmos no servidor.

Para avaliar a escalabilidade do sistema proposto, foram realizadas medidas de tempo de resposta na geração da recomendação para 1000 usuários selecionados aleatoriamente. A Figura 15 apresenta os tempos de processamento em milissegundos das recomendações por popularidade, baseada em conteúdo e colaborativa para estes 1000 usuários.

Figura 15 – Tempo de resposta do Primeiro Experimento



Fonte: Elaborada pelo autor

Em termos de tempo médio de resposta, os resultados apresentados na Figura 15 mostram-se promissores: 32,53 ms para recomendação

por popularidade (desvio padrão de 17,04); 164,76ms para a FBC (desvio padrão de 21,34); e 459,40ms para a FC (desvio padrão de 417,36). Em termos de média, o tempo de resposta das recomendações ficaram muito inferiores a 2s, considerado o limite tolerável de respostas do ponto de vista de usuários da Web (NAH, 2004).

Analisando apenas a recomendação por popularidade, observa-se que este tipo de recomendação tem um tempo de resposta bem inferior aos anteriores, inclusive em termos de desvio padrão. Isto é justificado pelo fato que a consulta ao banco de dados é simples, resultando em tempos de processamento menores. A menor variação se deve ao fato que a consulta é a mesma, independente do usuário.

Na recomendação por FBC, os tempos de resposta são muito superiores a recomendação por popularidade, devido ao fato da consulta ser mais complexa, requerendo maior tempo de processamento em nível do banco de dados Neo4J. Existe uma variabilidade maior, devido ao fato que os perfis dos usuários serem diferentes.

O pior cenário é quando do uso da recomendação por FC. Neste caso, as consultas no banco de dados exigem um maior poder de processamento. A grande variabilidade dos tempos de resposta neste caso se deve a diferentes números de vizinhos próximos do usuário foco da recomendação. Observa-se também que existem casos em que o tempo de resposta ultrapassou os 2s. Isto ocorreu para três usuários, que tiveram um grande número de vizinhos próximos. Observou-se neste tipo de recomendação o maior atraso, alcançando tempos de resposta de em torno de 3s. Isto demonstra que o protótipo desenvolvido deveria passar uma etapa de otimização do código e do gerenciador de banco de dados, inclusive otimizando das consultas ao banco de dados.

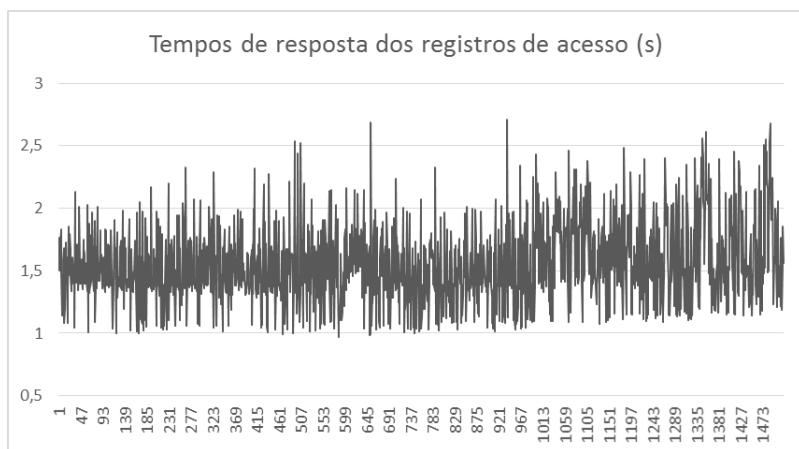
5.4.2 Avaliação usando a BDLB

Neste experimento, foi utilizada a implementação da BDLB utilizando o *Web Service* de recomendação proposto (apresentada na Seção 5.2). Para este experimento, a BDLB estava instalada em um computador com processador Intel i5 de 2.67GHz e com 8GB de memória, e o *Web Service* de recomendação em um servidor com processador Intel XEON E5404 de 2.0 GHz e 8GB de memória. As duas máquinas possuem sistema operacional Ubuntu versão 14.04 LTS, e estavam instaladas na rede campus da UFSC.

Inicialmente, foram simulados 1516 acessos de usuários a itens da BDLB. Estes acessos foram acessos reais realizados por 691 usuários

cadastrados na BDLB (usuários registrados que acessaram obras literárias). O gráfico da Figura 16 apresenta os tempos de resposta de cada uma destas chamadas *accessNotification*. Estes tempos incluem o atraso de processamento no cliente, latência da rede e tempo de processamento no servidor. Em termos de latência de rede, através de experimentos, observou-se que o atraso de ida-e-volta é inferior a 1ms. Portanto, ele não é considerado nesta análise. O tempo médio de processamento no cliente também é reduzido (1,17ms), podendo também ser desprezado.

Figura 16 – Tempos de resposta do registro de acesso



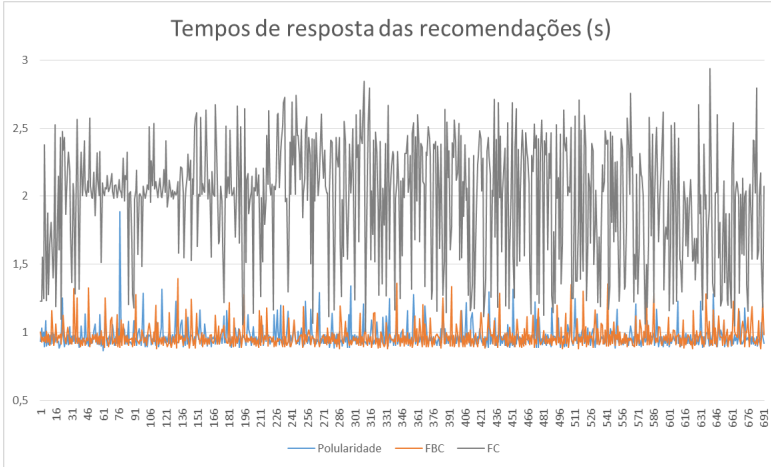
Fonte: Elaborada pelo autor

O tempo de resposta médio observado na chamada do método *accessNotification* foi de 1,55s, com desvio padrão de 0,31. Este tempo de resposta é completamente aceitável, pois ele não influencia na qualidade de navegação do usuário. Isto pois este tipo de registro é realizado em *background* no momento em que o usuário realiza um acesso a um item do RD. A variação dos tempos de resposta se deve principalmente a variação do número de autores e número de gêneros literários das obras acessadas. Isto implica no aumento ou redução dos relacionamentos ao nível da representação semântica, e consequente maior atraso no mapeamento para um grafo e no processamento em nível do banco de dados.

Após registrado todos os acessos a itens por parte dos usuários, foram simuladas 691 solicitações de recomendação, uma para cada

usuário considerado neste estudo de caso. A Figura 17 apresenta os tempos de processamento em segundos das recomendações por popularidade, baseada em conteúdo e colaborativa deste experimento.

Figura 17 – Tempos de resposta da solicitação da recomendação



Fonte: Elaborada pelo autor

Na recomendação por popularidade, o tempo de resposta médio foi de 0,972s, com desvio padrão de 0,084. É importante observar que na recomendação por popularidade, a recomendação é a mesma, independente do usuário. Esta pequena variação de valores se deve às variações de atraso na rede e nos tempos de processamentos tanto no cliente quanto no servidor.

No caso da FBC, o tempo de resposta médio foi de 0,973s, com desvio padrão de 0,087s. O tempo de resposta foi praticamente o mesmo para aquele com popularidade. Apesar da consulta FBC no banco de dados gerar um atraso maior, os tempos de consulta no banco de dados ficaram na ordem de milissegundos e o atraso maior se deve aos demais processamentos no lado do servidor.

Já na FC, os tempos de resposta foram bem superiores. O tempo de resposta médio foi de 1,98s, com desvio padrão de 0,42. Apesar do tempo de resposta ser muito superior aos dois tipos anteriores, ele ficou, em média, abaixo de 2s, não comprometendo a qualidade de navegação dos usuários. Houve também uma variação do valor de atraso maior, devido principalmente ao número de vizinhos próximos do usuário foco da recomendação e também o número de itens que estes vizinhos próximos

accessaram.

6 CONCLUSÃO

Muitas organizações não fazem uso de sistemas de recomendação em seus RDs devido provavelmente a inexistência de soluções que possam ser facilmente integradas ao RD, sem muito esforço de desenvolvimento. Para atender tal demanda, este trabalho propôs um sistema de recomendação fracamente acoplado com RDs e adaptável ao domínio dos conteúdos do RD, graças ao seu provimento via *Web Services* e uso de ontologias.

A técnica aqui proposta apresentou uma estrutura global de recomendação utilizando três conceitos principais: Fator de interesse, item e audiência. A partir destes itens e utilizando-se da estrutura de ontologias, criou-se uma modelagem que, com o auxílio de um *Web Service*, tornou possível o uso de um sistemas de recomendação pelos mais diversos RDs, com uma ampla possibilidade de associação de conteúdo, tendo em vista que os metadados de um item e suas relações podem ser descritas e persistidas por meio da ontologia.

Esta dissertação apresentou também alguns experimentos que mostraram a factibilidade da proposta, com tempos de processamento adequados para oferecer tempos de resposta nos limites aceitáveis para usuários da Web. Além disso, foi apresentado por meio de experimentos, a adaptabilidade da técnica a diferentes RDs em domínios distintos.

Com os testes realizados, a premissa inicial do trabalho, que pressupunha que um sistema de recomendação para RDs utilizando ontologias para representação de conteúdo poderia ser viável utilizando o armazenamento das informações em forma de grafos, otimizando assim o tempo de processamento e compensando eventuais atrasos de rede, foi demonstrada.

Os algoritmos utilizados e os ambientes de produção (BDLB e Dspace), apresentaram não só a factibilidade do modelo proposto como abriram diversos precedentes para trabalhos futuros. Com relação a factibilidade da proposta, observou-se que o fato do modelo proposto ser processado a partir de um *Web Service* não influenciou negativamente no tempo de resposta da recomendação, apresentando resultados aceitáveis, tendo em vista que o cliente não precisa implementar técnicas de recomendação.

A técnica aqui descrita ainda apresenta alguns pontos fracos, que podem ser explorados em trabalhos futuros. O primeiro é prover um menor impacto no lado do RD para que sejam monitorados os acessos dos usuários e a geração da recomendação. Para tanto, pretende-se

gerar *add-ons* para o DSpace e Okema, facilitando ainda mais a utilização do serviço de recomendação. Outra medida é incorporar um mecanismo permitindo a importação automática dos itens a recomendar e seus metadados utilizando o protocolo PMH-OAI (PMH-OAI, 2017), pois no modelo proposto a base de conhecimento é gerada a partir de cada acesso de usuário, ou seja, um item só pode ser recomendado se em algum momento um usuário o acessou.

Com relação a vertentes futuras que a proposta apresentada nesta dissertação pode tomar, é a continuação do desenvolvimento da Ontologia de Recomendação RecOnt. Outro trabalho futuro indicado é a implantação de novos algoritmos de recomendação dentro do modelo, podendo até mesmo utilizar o modelo como framework entre novas técnicas de recomendação. Pode-se também trabalhar-se a arquitetura apresentada para que o tempo de resposta de recomendação seja otimizado, com cálculos paralisados.

Por fim, a técnica proposta abre precedentes para ser utilizada em Federações de RDs, uma vez que a RecOnt padroniza os conteúdos a serem acessados e o *Web Service* separa os conteúdos por clientes e por contextos. Uma arquitetura *top-down* pode ser proposta para abrigar uma federação de RDs.

Este trabalho resultou em uma publicação científica intitulada “Recommending Web Service Based on Ontologies for Digital Repositories” no WebMedia 2015 - XXI Simpósio Brasileiro de Multimídia e Web. Este evento possui Estrato Qualis/CAPES B3.

REFERÊNCIAS

- ACM, D. L. **Standard for Learning Object Metadata**. 2017. [Http://libraries.acm.org/digital-library](http://libraries.acm.org/digital-library). Accessed: 2017-02-02.
- ANJORIN, M. et al. A framework for cross-platform graph-based recommendations for tel. In: **Proceedings of the 2nd workshop on recommender systems in technology enhanced learning**. [S.l.: s.n.], 2012. p. 83–88.
- BALABANOVIC, M.; SHOHAM, Y. Fab: Content-based, collaborative recommendation. **Comm. ACM**, vol. 40, no. 3, pp. 66-72, 1997.
- BATRA, S.; TYAGI, C. Comparative analysis of relational and graph databases. **International Journal of Soft Computing and Engineering**, 2012.
- BEHAM, G. et al. Recommending knowledgeable people in a work-integrated learning system. **Procedia Computer Science**, Elsevier, v. 1, n. 2, p. 2783–2792, 2010.
- BORST, W. N. **Construction of Engineering Ontologies for Knowledge Sharing and Reuse**. Tese (Doutorado), Enschede, September 1997. Disponível em: <<http://doc.utwente.nl/17864/>>.
- BOUTEMEDJET, S.; ZIOU, D. A graphical model for context-aware visual content recommendation. **IEEE Transactions on Multimedia**, 10(1), p. 52?62, 2008.
- BRIN, S.; PAGE, L. Reprint of: The anatomy of a large-scale hypertextual web search engine. **Computer networks**, Elsevier, v. 56, n. 18, p. 3825–3833, 2012.
- BURKE, R. Hybrid web recommender systems. **The Adaptive Web: Methods and Strategies of Web Personalization**. LNCS, vol. 4321, pp. 377?408, 2007.
- CANTADOR, I.; CASTELLS, P.; BELLOGÍN, A. An enhanced semantic layer for hybrid recommender systems: Application to news recommendation. **International Journal on Semantic Web and Information Systems (IJSWIS)**, IGI Global, v. 7, n. 1, p. 44–78, 2011.

CASAGRANDE, M. F. R.; KOZIMA, G.; WILLRICH, R. A recommendation technique based on metadata for digital repositories oriented to learning. **Brazilian Journal of Computers in Education**, v. 23, n. 02, p. 70, 2015.

CASTAGNÉ, M. **Institutional repository software comparison: DSpace, EPrints, Digital Commons, Islandora and Hydra**. 2013.

<https://open.library.ubc.ca/cIRcle/collections/42591/items/1.0075768>. Accessed: 2017-01-09.

CAZELLA, S. C.; NUNES, M. Ciência da opinião; estado da arte em sistemas de recomendação. **XXX Congresso da SBC**, 2010.

CLAYPOOL, M. e. a. Combining content-based and collaborative filters in an online newspaper. **Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation**, 1999.

COMMONS, D. **The leading hosted institutional repository software for universities, colleges, law schools, and research centers**. 2016. <http://digitalcommons.bepress.com/>. Accessed: 2017-01-10.

CROW, R. The case for institutional repositories: A sparcs position paper. **Scholarly Publishing e Academic Resources Coalition**, 2002.

DANER, C. The collaborative filtering recommendation algorithm based on bp neural networks. **Proceedings of the International Symposium on Intelligent Ubiquitous Computing and Education**, p. 234-236, 2009.

DEMOVIC, L. et al. Movie recommendation based on graph traversal algorithms. In: **DEXA Workshops**. [S.l.: s.n.], 2013. p. 152-156.

D.H., P. et al. A literature review and classification of recommender systems research. **Expert Systems with Applications**, **39** (11), p. 10059-10072, 2012.

DSPACE. **DSpace - DSpace is a turnkey institutional repository application**. 2015. <http://www.dspace.org/>. Accessed: 2015-06-26.

DUNCAN, C. Repositories: e-learning for everyone. **eLearnInternational, Edinburgh 9-12**, 2003.

EASYREC. **EasyRec, open source recommendation engine**. 2017. [Http://easyrec.org/](http://easyrec.org/). Accessed: 2017-01-10.

EKSTRAND, M. D. et al. Rethinking the recommender research ecosystem: Reproducibility, openness, and lenskit. In: **Proceedings of the Fifth ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2011. (RecSys '11), p. 133–140. ISBN 978-1-4503-0683-6. Disponível em: <<http://doi.acm.org/10.1145/2043932.2043958>>.

ELLIOTT, D.; RUTHERFORD, J.; ERICKSON, J. A **recommender system for the DSpace open repository platform**. [S.l.], 2008.

ENGELKE, B. V.; WILLRICH, R. **Sistema de recomendação para a plataforma DSpace**. [S.l.], 2016.

EPRINTS. **world-leading open-source digital repository platform**. 2016. [Http://www.eprints.org/](http://www.eprints.org/). Accessed: 2017-01-10.

FEDORA. **Fedora Repository**. 2017. [Http://fedorarepository.org/](http://fedorarepository.org/). Accessed: 2017-01-02.

FURTADO, C. et al. Ordenação personalizada na recuperação de informações em bibliotecas digitais. **Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)**, 2009.

GANTNER, Z. et al. Mymedialite: A free recommender system library. In: **Proceedings of the Fifth ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2011. (RecSys '11), p. 305–308. ISBN 978-1-4503-0683-6. Disponível em: <<http://doi.acm.org/10.1145/2043932.2043989>>.

GOLDBERG, D. et al. Using collaborative filtering to weave an information tapestry. **Communications of the ACM** **35**, 1992.

GOLDBERG, D.; NICHOLSL, D. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, **V. 35**, **N.12**, **P 61-70**, 1992.

GROUP, W. O. W. **Web Services Architecture**. 2004. [Https://www.w3.org/TR/ws-arch/](https://www.w3.org/TR/ws-arch/). Accessed: 2017-01-09.

GROUP, W. O. W. **OWL 2 Web Ontology Language Document Overview (Second Edition)**. 2012. [Http://www.w3.org/TR/owl2-overview/](http://www.w3.org/TR/owl2-overview/). Accessed: 2015-06-26.

GUARINO, N. Formal ontology and information systems.
Proceedings of FOIS, v. 98, n. 1998, p. 3–15, 1998.

HEERY, R.; ANDERSON, S. **Digital repositories review**. 2005.
[Http://www.jisc.ac.uk/uploaded_documents/digitalrepositoriesreview-2005.pdf](http://www.jisc.ac.uk/uploaded_documents/digitalrepositoriesreview-2005.pdf). Accessed: 2015-06-26.

HORRIDGE, M.; BECHHOFFER, S. The owl api: A java api for owl ontologies. **Semantic Web**, v. 2, n. 1, p. 11–21, 2011.

HUANG, Q. et al. Personalized video recommendation through graph propagation. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, New York, NY, USA, v. 10, n. 4, p. 32:1–32:17, jul. 2014. ISSN 1551-6857. Disponível em:
 <<http://doi.acm.org/10.1145/2598779>>.

HUANG, Z.; LI, X.; CHEN, H. Link prediction approach to collaborative filtering. **Proceedings of the 5th ACM/IEEECS joint conference on Digital libraries**, 2005.

IEEE. **IEEE Digital Library**. 2017.
[Http://ieeexplore.ieee.org/Xplore/home.jsp](http://ieeexplore.ieee.org/Xplore/home.jsp). Accessed: 2017-02-02.

IEEE-LTSC. **IEEE LTSC Working Group 12: Learning Object Metadata**. 2004. [Http://ltsc.ieee.org/wg12/index.html](http://ltsc.ieee.org/wg12/index.html). Acesso em janeiro. 2017.

IMS, G. L. C. **Standard for Learning Object Metadata**. 2002.
[Https://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html](https://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html). Accessed: 2017-02-02.

INITIATIVE, D. C. M. **Dublin Core Metadata Element Set, Versão 1.1**. 2014. [Http://dublincore.org/documents/dces/](http://dublincore.org/documents/dces/). Acesso em janeiro. 2017.

JACOBSON, K.; FERRIS, B. **The Recommendation Ontology 0.3: Namespace Document 02 August 2010**. 2010.
[Http://smiy.sourceforge.net/rec/spec/recommendationontology.html](http://smiy.sourceforge.net/rec/spec/recommendationontology.html). Accessed: 2015-06-26.

KANTOR, P. B. et al. **Recommender systems handbook**. [S.l.]: Springer, 2011.

LEMOS, F. D. et al. Improving photo recommendation with context awareness. In: **Proceedings of the 18th Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2012. (WebMedia '12), p. 321–330. ISBN 978-1-4503-1706-1. Disponível em: <<http://doi.acm.org/10.1145/2382636.2382704>>.

MIDDLETON, S.; SHADBOLT, N.; ROURE, D. Ontological user profiling in recommender systems. **Information Systems**, vol. **22**, no. **1**, pp. **54-88**, 2004.

MOONEY, R.; ROY, L. Content-based book recommending using learning for text categorization. **Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation**, 1999.

NAH, F. F.-H. A study on tolerable waiting time: how long are web users willing to wait? **Behaviour & Information Technology**, v. 23, n. 3, p. 153–163, 2004. Disponível em: <<http://dx.doi.org/10.1080/01449290410001669914>>.

NEO4J. **Neo4j, the World's Leading Graph Database**. 2017. [Http://www.neo4j.com](http://www.neo4j.com). Accessed: 2015-06-26.

NISO, N. I. S. O. **Understanding Metadata**. 2004. [Http://www.niso.org/publications/press/UnderstandingMetadata.pdf](http://www.niso.org/publications/press/UnderstandingMetadata.pdf). Accessed: 2017-01-10.

ODONOVAN, J. et al. Peerchooser: visual interactive recommendation. **Proceedings of the 26th SIGCHI Conference on Human Factors in Computing Systems (CHI'08)**, ACM Press, New York, NY, USA. p. 1085?1088, 2008.

OLIVEIRA, R. R. d.; CARVALHO, C. L. **bibliotecas digitais e o repositório fedora**. 2011. Accessed: 2017-01-22.

OMEKA. **Omeka Homepage**. 2017. [Http://www.Omeka.org/](http://www.Omeka.org/). Accessed: 2015-06-26.

OWEN, S. et al. **Mahout in Action**. Greenwich, CT, USA: Manning Publications Co., 2011. ISBN 1935182684, 9781935182689.

PMH-OAI, O. A. I. **Open Archives Initiative Protocol for Metadata Harvesting**. 2017. [Https://www.openarchives.org/pmh/](https://www.openarchives.org/pmh/). Accessed: 2015-06-26.

PROTÉGÉ. **Protégé**. 2017. [Http://protege.stanford.edu](http://protege.stanford.edu). Accessed: 2017-02-02.

REGO, P. A. et al. Mapreduce performance evaluation for knowledge-based recommendation of context-tagged photos. In: **Proceedings of the 19th Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2013. (WebMedia '13), p. 249–256. ISBN 978-1-4503-2559-2. Disponível em: <<http://doi.acm.org/10.1145/2526188.2530537>>.

RESNICK, P.; VARIAN, H. R. Recommender systems. **Communications of the ACM**. v.40,n.3,p.55-58, 1997.

SAYÃO, L. S. F. Padrões para bibliotecas digitais abertas e interoperáveis. **Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação**, p. 18–47, 2007.

SCIENCEDIRECT. **Science Direct**. 2017. <https://blog.sciencedirect.com/topics/sciencedirect>. Accessed: 2017-02-02.

SIEG, A.; MOBASHER, B.; BURKE, R. Improving the effectiveness of collaborative recommendation with ontology-based user profiles. In: **ACM. proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems**. [S.l.], 2010. p. 39–46.

SMITH, M. et al. **Dspace: An open source dynamic digital repository**. 2003.

TATIYA, R. V.; VAIDYA, A. S. A survey of recommendation algorithms. **Journal of Computer Engineering**, 2014.

TORRES, R. et al. Enhancing digital libraries with techlens+. **Proceedings of the 2004 joint ACM/IEEE conference on Digital libraries Pages: 228 - 236**, 2004.

VELLINO, A.; ZEBER, D. A. H. Multi-dimensional recommender for journal articles in a scientific digital library. **Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops**, p.111-114, 2007.

VIANA, C. L. de M.; ARELLANO, M. Ángel M.; SHINTAKU, M. Repositórios institucionais em ciência e tecnologia : uma experiência

de customização do dspace. **Simposio Internacional de Bibliotecas Digitais**, 2005.

WAN G. E LIU, Z. Content-based information retrieval and digital libraries. **Information Technology and Libraries** **27**, 41?47, 2000.

WILLRICH, R. et al. Adaptive information retrieval system applied to digital libraries. In: **Proceedings of the 12th Brazilian Symposium on Multimedia and the Web**. New York, NY, USA: ACM, 2006. (WebMedia '06), p. 165–173. ISBN 85-7669-100-0. Disponível em: <<http://doi.acm.org/10.1145/1186595.1186616>>.

WOLF, A. S.; MONTEIRO, A. P. L.; VALMORBIDA, W. **Biblioteca digital da univates utilizando o software dspace**. 2013. Destaques Acadêmicos 1 (4).

ANEXO A – Ontologia RecOnt

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rdf:RDF xmlns="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:RecOnt="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#"
xml:base="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl">
  <owl:Ontology rdf:about="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl"/>
  <!--
    //////////////////////////////////////
    //
    // Object Properties
    //
    //////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#accessed
-->
<owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#accessed">
  <rdfs:domain rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Audience"/>
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Item"/>
</owl:ObjectProperty>
<!--
  http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#hasFI
-->
<owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#hasFI">
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
  <rdfs:domain rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Item"/>
</owl:ObjectProperty>
<!--

    //////////////////////////////////////
    //
    // Classes
    //
    //////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Audience
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Audience"/>
<!--
  http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
<!--
  http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Item
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Item"/>
</rdf:RDF>
<!--
  Generated by the OWL API (version 3.5.1) http://owlapi.sourceforge.net
-->
```


ANEXO B – Ontologia do Domínio de Literatura

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rdf:RDF xmlns="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://biblio.inf.ufsc.br/~anderson/Literatura.owl">
  <owl:Ontology rdf:about="http://biblio.inf.ufsc.br/~anderson/Literatura.owl"/>
    <!--

    //////////////////////////////////////
    //
    // Object Properties
    //
    //////////////////////////////////////

-->
    <!--
      http://biblio.inf.ufsc.br/~anderson/Literatura.owl#hasAuthor
    -->
    <owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#hasAuthor">
      <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Author"/>
      <rdfs:domain
        rdf:resource="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Literary_work"/>
    </owl:ObjectProperty>
    <!--
      http://biblio.inf.ufsc.br/~anderson/Literatura.owl#hasGenre
    -->
    <owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#hasGenre">
      <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Genre"/>
      <rdfs:domain
        rdf:resource="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Literary_work"/>
    </owl:ObjectProperty>
    <!--

    //////////////////////////////////////
    //
    // Classes
    //
    //////////////////////////////////////

-->
    <!--
      http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Author
    -->
    <owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Author"/>
    <!--
      http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Genre
    -->
    <owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Genre"/>
    <!--
      http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Literary_work
    -->
    <owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/Literatura.owl#Literary_work"/>
  </rdf:RDF>
  <!--
    Generated by the OWL API (version 3.5.1) http://owlapi.sourceforge.net
  -->
```


ANEXO C – Ontologia de Contexto de Literatura

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rdf:RDF xmlns="http://biblio.inf.ufsc.br/~anderson/Literature.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://biblio.inf.ufsc.br/~anderson/Literature.owl">
  <owl:Ontology rdf:about="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl">
    <owl:imports rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl"/>
  </owl:Ontology>
<!--

////////////////////////////////////
//
// Object Properties
//
////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#hasAuthor
-->
<owl:ObjectProperty
  rdf:about="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#hasAuthor">
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Author"/>
  <rdfs:domain
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Literary_work"/>
</owl:ObjectProperty>
<!--
  http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#hasGenre
-->
<owl:ObjectProperty
  rdf:about="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#hasGenre">
  <rdfs:domain
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Literary_work"/>
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Genre"/>
</owl:ObjectProperty>
<!--

////////////////////////////////////
//
// Classes
//
////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Author
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Author">
  <rdfs:subClassOf
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
</owl:Class>
<!--
  http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Resource
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Literary_work">
  <rdfs:subClassOf rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Item"/>
</owl:Class>
<!--
  http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Genre
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/LiteratureContext.owl#Genre">
  <rdfs:subClassOf
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
</owl:Class>
</rdf:RDF>
```

```
<!--  
  Generated by the OWL API (version 3.5.1) http://owlapi.sourceforge.net  
-->
```


ANEXO D – Ontologia de Domínio do Experimento DSpace

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rdf:RDF xmlns="http://biblio.inf.ufsc.br/~anderson/DCResource.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://biblio.inf.ufsc.br/~anderson/DCResource.owl">
  <owl:Ontology rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl">
    <owl:imports rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl"/>
  </owl:Ontology>
<!--

////////////////////////////////////
//
// Object Properties
//
////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasAuthor
-->
<owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasAuthor">
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Author"/>
  <rdfs:domain rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource"/>
</owl:ObjectProperty>
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasSubject
-->
<owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasSubject">
  <rdfs:domain rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource"/>
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Subject"/>
</owl:ObjectProperty>
<!--

////////////////////////////////////
//
// Classes
//
////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Author
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Author">
  <rdfs:subClassOf
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
</owl:Class>
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource">
  <rdfs:subClassOf rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Item"/>
</owl:Class>
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Subject
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Subject">
  <rdfs:subClassOf
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
</owl:Class>
</rdf:RDF>
<!--
  Generated by the OWL API (version 3.5.1) http://owlapi.sourceforge.net
-->
```


ANEXO E – Ontologia de Contexto do Experimento DSpace

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rdf:RDF xmlns="http://biblio.inf.ufsc.br/~anderson/DCResource.owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://biblio.inf.ufsc.br/~anderson/DCResource.owl">
  <owl:Ontology rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl">
    <owl:imports rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl"/>
  </owl:Ontology>
<!--

////////////////////////////////////
//
// Object Properties
//
////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasAuthor
-->
<owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasAuthor">
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Author"/>
  <rdfs:domain rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource"/>
</owl:ObjectProperty>
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasSubject
-->
<owl:ObjectProperty rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#hasSubject">
  <rdfs:domain rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource"/>
  <rdfs:range rdf:resource="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Subject"/>
</owl:ObjectProperty>
<!--

////////////////////////////////////
//
// Classes
//
////////////////////////////////////

-->
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Author
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Author">
  <rdfs:subClassOf
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
</owl:Class>
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Resource">
  <rdfs:subClassOf rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Item"/>
</owl:Class>
<!--
  http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Subject
-->
<owl:Class rdf:about="http://biblio.inf.ufsc.br/~anderson/DCContext.owl#Subject">
  <rdfs:subClassOf
    rdf:resource="http://biblio.inf.ufsc.br/~anderson/RecOnt.owl#Fator_of_interest"/>
</owl:Class>
</rdf:RDF>
<!--
  Generated by the OWL API (version 3.5.1) http://owlapi.sourceforge.net
-->
```